



UNIVERISTE MOHAMMED PREMIER
École Nationale des Sciences Appliquées
Oujda

جامعة محمد الأول
المدرسة الوطنية للعلوم التطبيقية وجدة

Mémoire de Projet de Fin d'année

Spécialité : Sécurité Informatique & Cyber-Sécurité

Présenté par :

RAGGUI Salah Eddine & FILALI Abdelilah & MOUAATARIF Anass

Sujet intitulé:

**La mise en place d'une solution open source
SIEM/XDR**

Membres de jury:

M. SEFRAOUI Omar

M. MADANI Mohamed Amine

Encadrés par:

M. SEFRAOUI Omar

Année universitaire 2022-2023

Remerciements

Nous exprimons notre gratitude envers Dieu, qui nous a donné la force et la détermination nécessaires pour surmonter les difficultés rencontrées et mener ce travail à terme.

Un grand merci à notre Professeur, Monsieur OMAR SEFRAOUI, de l'école nationale des sciences appliquées d'Oujda, pour son encadrement précieux. Ses conseils avisés et son soutien constant ont été d'une grande aide tout au long de la réalisation de ce mémoire.

Nous souhaitons également remercier sincèrement les membres du jury qui ont évalué notre travail avec attention et objectivité.

Enfin, nous tenons à exprimer notre profonde reconnaissance envers toutes les personnes qui ont contribué à la réussite de ce projet. Votre soutien, vos conseils et votre participation ont été essentiels et nous sommes extrêmement reconnaissants de votre apport.

Nous tenons à vous remercier du fond du cœur pour votre contribution inestimable. Cette réussite n'aurait pas été possible sans votre aide et nous vous en sommes extrêmement reconnaissants

Résumé

Ce projet avait pour but d'améliorer notre solution SIEM en mettant l'accent sur la personnalisation des règles de détection et la mise en place d'une réponse active aux incidents de sécurité. En utilisant la solution open source **Wazuh**, nous avons renforcé notre capacité à détecter les menaces spécifiques à notre environnement et à réagir de manière plus efficace aux incidents de sécurité.

La personnalisation des règles de détection nous a permis d'obtenir des alertes plus précises, tandis que la réponse active a permis de prendre des mesures immédiates pour arrêter les attaques en cours, bloquer les adresses IP suspectes et isoler les machines compromises. Grâce à ces améliorations, notre solution SIEM est devenue plus robuste, renforçant ainsi notre posture de sécurité globale contre les cybermenaces.

TABLE DE MATIERES

Abstract	7
Introduction générale	8
Chapitre 1: Présentation général	9
1.1. Introduction à la solution SIEM/XDR	9
1.2. Types d'attaques utilisées dans le projet	11
1.3. Outils d'exploitation utilisés dans le projet.....	12
1.5. Les composants de Wazuh	13
Chapitre 2: Environnement Wazuh	21
2.1. Introduction	21
2.2. Environnement de déploiement	21
2.3. Installation du serveur Wazuh	22
2.4. Installation des clients Wazuh	23
2.5. Connectivité des clients avec le serveur	26
2.6. Test des règles par défaut	27
Chapitre 3: Renforcement de la sécurité avec Wazuh.....	32
3.2. Active Response	35
3.3. VirusTotal.....	37
3.4. Intégrité des fichiers	40
3.5. Détection des attaques par injection SQL	41
Conclusion Générale	43
Bibliographie.....	44

Table des figures

<i>Figure 1: Composants de Wazuh</i>	<i>14</i>
<i>Figure 2 : Architecture de l'indexeur.....</i>	<i>15</i>
<i>Figure 3 : Architecture du serveur</i>	<i>16</i>
<i>Figure 4 : Dashboard Wazuh</i>	<i>18</i>
<i>Figure 5 : Architecture du client</i>	<i>19</i>
<i>Figure 6 : Schéma de déploiement</i>	<i>21</i>
<i>Figure 7: Installation de la clé GPG.....</i>	<i>24</i>
<i>Figure 8 : Ajout du dépôt</i>	<i>24</i>
<i>Figure 9 : Mis à jour des packages.....</i>	<i>24</i>
<i>Figure 10 : Installation du client "Linux"</i>	<i>25</i>
<i>Figure 11 : Activation et Démarrage du service agent</i>	<i>25</i>
<i>Figure 12 : Vérification du status Wazuh</i>	<i>25</i>
<i>Figure 13 : Déploiement du client "Windows"</i>	<i>26</i>
<i>Figure 14 : Configuration du client</i>	<i>26</i>
<i>Figure 15 : Résultat de connectivité.....</i>	<i>27</i>
<i>Figure 16 : Détection des connections SSH.....</i>	<i>28</i>
<i>Figure 17 : Détection d' escalade de privilèges</i>	<i>28</i>
<i>Figure18 : Détection des bruteforce</i>	<i>29</i>
<i>Figure19 : Détection de Hydra bruteforce</i>	<i>29</i>
<i>Figure 20 : Détection de scan NMAP</i>	<i>30</i>
<i>Figure 21: Détection des Rootkits.....</i>	<i>31</i>
<i>Figure 22 : Détection De l'installation d'un nouveau package</i>	<i>31</i>
<i>Figure 23: Détection de vulnérabilités</i>	<i>31</i>
<i>Figure 24 : Règles de négociation SSH</i>	<i>33</i>
<i>Figure 25 : Décodeur associé.....</i>	<i>34</i>
<i>Figure 26: Résultat de la règle de négociation SSH</i>	<i>34</i>
<i>Figure 27 : Configuration de l'active response</i>	<i>35</i>
<i>Figure 28 : Succès du scan NMAP</i>	<i>35</i>
<i>Figure 29 : Détection du scan NMAP</i>	<i>36</i>

Figure 30 : Echec du scan nmap.....	36
Figure 31 : Echec des ping	36
Figure 32 : Configuration de la surveillance.....	37
Figure 33 : Installation jq	37
Figure 34 :Script pour suppression des fichiers malveillants	37
Figure 35 : Modification des permissions.....	38
Figure 36 : Redemarrage Coté Serveur	38
Figure 37 : Règle d'alerte	38
Figure 38 : Règles de reception d'alerte	38
Figure 39 : Integration de VirusTotal	39
Figure 40 : Activation de réponse	39
Figure 41 : Téléchargement du virus.....	39
Figure 44 : Activation de la surveillance d'intégrité des fichiers	40
Figure 45 : Modification du fichier file.txt.....	40
Figure 46 : Résultat d'alert après la modification	41
Figure 47 : Configuration de la surveillance des fichiers log "web apache".....	41
Figure 48 : Injection SQL	42
Figure 49 : Erreur 404	42

Abstract

The abstract highlights the significance of SIEM and XDR solutions in addressing the growing cyber threats faced by organizations. SIEM provides real-time security event monitoring and correlation, while XDR extends threat detection and response capabilities by integrating data from multiple sources.

SIEM systems analyze vast amounts of data to detect security incidents and abnormal activities, offering proactive alerts and response. However, traditional SIEM solutions have limitations, which is where XDR comes into play. XDR integrates diverse security tools to detect cross-environment threats that individual tools may miss.

By combining SIEM and XDR, organizations benefit from centralized visibility, advanced threat intelligence, and real-time detection and response capabilities. Implementing SIEM/XDR solutions leads to improved threat detection accuracy, reduced response time, operational efficiency, and compliance with industry regulations.

However, organizations face challenges such as data integration and scalability, requiring qualified cybersecurity personnel to effectively implement these solutions.

In conclusion, SIEM/XDR solutions provide a holistic and proactive approach to cybersecurity, enabling organizations to stay ahead of evolving threats, protect critical assets, and maintain a strong security posture in today's complex digital landscape.

Introduction générale

Dans les environnements informatiques d'aujourd'hui, la sécurité des données et la protection contre les menaces sont des préoccupations majeures pour les organisations. Les attaques informatiques sont de plus en plus sophistiquées, et il est essentiel de disposer d'une solution de surveillance et de détection des intrusions robuste.

Ce projet se concentre sur l'implémentation d'une solution SIEM (Security Information and Event Management) et XDR (Extended Detection and Response) open source avec Wazuh. Wazuh est une plateforme de sécurité qui offre des fonctionnalités avancées de détection d'intrusion, de gestion des journaux et de conformité réglementaire pour les environnements informatiques.

L'objectif de ce projet est de renforcer la sécurité des systèmes en mettant en place une solution complète de surveillance et de détection des menaces. Nous utilisons Wazuh pour collecter et analyser les journaux des événements système, les fichiers de configuration et les informations de sécurité, afin de détecter les comportements malveillants et les anomalies.

Dans ce rapport, nous présenterons la méthodologie utilisée pour implémenter la solution SIEM et XDR avec Wazuh. Nous décrirons en détail les étapes de déploiement, de configuration et d'intégration de Wazuh dans l'environnement informatique. Nous mettrons l'accent sur les fonctionnalités clés de Wazuh, telles que la détection d'intrusion, la corrélation d'événements, la gestion des journaux et la conformité réglementaire.

Enfin, nous évaluerons l'efficacité de la solution mise en place en testant des attaques sur les agents surveillés par Wazuh. Nous analyserons les résultats obtenus et discuterons des améliorations potentielles à apporter à la solution pour renforcer davantage la sécurité des systèmes.

Chapitre 1 : Présentation général

1.1. Introduction à la solution SIEM/XDR

a. Historique des solutions SIEM/XDR :

Le domaine des solutions SIEM (Security Information and Event Management) et XDR (Extended Detection and Response) a connu une évolution significative au cours des deux dernières décennies. Les premières solutions SIEM ont émergé dans les années 2000, axées sur la gestion des événements de sécurité et la corrélation des logs pour détecter les menaces. Au fil du temps, ces solutions se sont améliorées pour intégrer davantage de sources de données et offrir une visibilité plus étendue.

Le concept d'XDR a commencé à prendre forme vers le milieu des années 2010, en réponse à l'évolution du paysage des menaces. Les premières solutions XDR ont élargi la portée de la détection des menaces en incluant des sources de données supplémentaires telles que les endpoints, les réseaux et les charges de travail dans le cloud. Cette approche plus holistique a permis une corrélation plus précise des événements et une meilleure compréhension des attaques.

Depuis lors, les solutions SIEM/XDR ont continué à évoluer rapidement. Au cours des dernières années, notamment entre 2018 et 2020, nous avons assisté à une adoption croissante des capacités d'analyse comportementale, d'apprentissage automatique et d'automatisation des réponses. Ces avancées ont permis de renforcer les capacités de détection précoce, de réduire les faux positifs et de permettre des réponses plus rapides et plus efficaces aux incidents de sécurité.

Aujourd'hui, les solutions SIEM/XDR sont devenues des composants essentiels de la stratégie de sécurité des organisations. Elles continuent d'évoluer pour s'adapter aux nouvelles menaces et aux besoins des entreprises, avec des innovations constantes dans les domaines de l'intelligence artificielle, de l'automatisation et de l'intégration avec d'autres solutions de sécurité.

b. Cinq solutions populaires SIEM/XDR:

Splunk Enterprise Security : Splunk est un acteur majeur dans le domaine de la gestion de la sécurité et offre une solution SIEM complète. Sa plateforme Enterprise Security combine des fonctionnalités SIEM avancées avec une visibilité en temps réel, la détection d'anomalies et la réponse aux incidents.

IBM QRadar: QRadar de IBM est une autre solution SIEM populaire, offrant une surveillance des événements en temps réel, une détection des menaces et une réponse automatisée. Il intègre également des capacités XDR pour une meilleure visibilité des attaques et une réponse coordonnée.

Wazuh : Wazuh est une plateforme open source de détection des menaces et de gestion de la conformité. Elle offre des fonctionnalités SIEM avancées, notamment la collecte des journaux, la détection d'anomalies, l'intégrité des fichiers, la détection des intrusions et la gestion des incidents.

Elastic Security : Elastic Security, basé sur la pile ELK (Elasticsearch, Logstash, Kibana), est une solution open source de SIEM et XDR. Il offre une surveillance des événements, une analyse des journaux, une détection des menaces et une réponse aux incidents.

OSSIM (Open-Source Security Information Management): OSSIM est une solution SIEM open source développée par AlienVault. Elle offre la collecte et la corrélation des événements de sécurité, la détection des menaces, la gestion des journaux et des rapports de sécurité.

c. Dix raisons pour choisir Wazuh pour votre environnement de sécurité:

1. **Détection des menaces avancée** : Wazuh offre des fonctionnalités puissantes de détection des menaces, y compris la surveillance des fichiers, la détection d'anomalies, la détection des intrusions et la corrélation des événements. Il vous permet de détecter rapidement les activités suspectes et les attaques potentielles.
2. **Collecte et analyse des logs** : Wazuh est capable de collecter et d'analyser les logs à partir de différents types de sources, tels que les systèmes d'exploitation, les applications, les pare-feux, les serveurs web, etc. Cela vous permet d'avoir une visibilité complète sur les événements de votre environnement.
3. **Intégration avec ELK Stack** : Wazuh s'intègre étroitement avec ELK Stack (Elasticsearch, Logstash, Kibana), qui est une pile open source populaire pour la collecte, la gestion et l'analyse des logs. Cette intégration vous offre des fonctionnalités avancées de recherche, de visualisation et de reporting des données de sécurité.
4. **Gestion des vulnérabilités** : Wazuh intègre également des fonctionnalités de gestion des vulnérabilités, ce qui vous permet de surveiller et de gérer les vulnérabilités présentes dans votre environnement. Cela vous aide à identifier les failles de sécurité potentielles et à prendre des mesures pour les résoudre.
5. **Conformité réglementaire** : Wazuh comprend des fonctionnalités spécifiques pour aider à répondre aux exigences de conformité réglementaire telles que PCI DSS, GDPR, HIPAA, etc. Il facilite la collecte des données nécessaires, l'analyse des événements de sécurité et la génération de rapports conformes aux normes.
6. **Réponse aux incidents** : Wazuh vous permet de configurer des règles personnalisées pour la détection d'incidents et de mettre en place des workflows de réponse automatisés. Cela vous aide à réagir rapidement aux incidents de sécurité, à minimiser les dommages potentiels et à contenir les attaques.

7. Intelligence des menaces : Wazuh est alimenté par une intelligence des menaces constamment mise à jour, qui vous permet de bénéficier d'une détection plus précise des nouvelles attaques et des techniques de piratage émergentes.
8. Facilité de déploiement : Wazuh est facile à déployer, que ce soit sur des machines physiques, virtuelles ou dans des environnements cloud. Il propose également des modules d'intégration prêts à l'emploi pour la collecte des logs à partir de différents systèmes.
9. Communauté active : Wazuh bénéficie d'une communauté active et engagée d'utilisateurs et de contributeurs. Vous pouvez trouver un support technique, des conseils et des ressources supplémentaires grâce à cette communauté.
10. Open source et évolutivité : Wazuh est une solution open source, ce qui signifie que vous avez accès au code source et pouvez le personnaliser selon vos besoins. Il est également évolutif, vous permettant de l'adapter à des environnements de toutes tailles, des petites entreprises aux grandes infrastructures.

1.2. Types d'attaques utilisées dans le projet

Dans le cadre de ce projet, différentes attaques sont utilisées pour évaluer la sécurité des systèmes et mettre en évidence les vulnérabilités potentielles. Parmi ces attaques, on trouve :

Privilège d'escalade (Privilege Escalation):



L'attaque de privilège d'escalade vise à augmenter les privilèges d'un utilisateur ou d'un processus dans un système. Les attaquants cherchent des vulnérabilités ou des faiblesses qui leur permettent de passer d'un niveau de privilège limité à un niveau de privilège plus élevé, leur donnant ainsi un accès plus étendu au système.

Attaque par force brute :



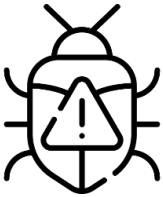
Une attaque par force brute est une méthode utilisée par les attaquants pour tenter de compromettre un système en essayant différentes combinaisons de mots de passe ou de clés jusqu'à ce qu'ils trouvent la bonne. Cela peut être utilisé pour accéder à des comptes utilisateur, des systèmes, des services ou même des fichiers chiffrés.

L'injection SQL :



Une technique d'attaque courante utilisée pour exploiter les vulnérabilités de sécurité dans les applications web qui interagissent avec une base de données SQL. L'attaque se produit lorsque des données non fiables sont insérées de manière incorrecte dans des requêtes SQL, ce qui permet à un attaquant d'exécuter des instructions SQL non autorisées.

Les malwares, ou logiciels malveillants :



Sont des programmes conçus dans le but de nuire à un système informatique, à des données ou à des utilisateurs. Ils peuvent prendre différentes formes, telles que des virus, des vers, des chevaux de Troie, des ransomwares, des spywares, etc.

Les malwares sont généralement distribués par le biais de techniques d'infection telles que des pièces jointes d'e-mails, des téléchargements de fichiers compromis, des liens malveillants, des sites web compromis, etc. Une fois qu'un malware infecte un système, il peut réaliser différentes actions nuisibles, telles que voler des informations confidentielles, endommager des fichiers, prendre le contrôle du système, ou encore demander une rançon pour restaurer l'accès aux données.

1.3. Outils d'exploitation utilisés dans le projet

Dans le cadre de ce projet, différents outils d'exploitation sont utilisés pour mener des attaques contrôlées et évaluer la sécurité des systèmes. Ces outils permettent aux professionnels de la sécurité de tester les vulnérabilités et d'identifier les points faibles dans les environnements informatiques. Parmi les outils couramment utilisés dans ce contexte, on trouve :

Nmap (Network Mapper) :



Un puissant outil open source de balayage de réseau utilisé pour l'exploration de réseau et l'audit de sécurité. Il vous permet de découvrir les hôtes d'un réseau, d'identifier les ports ouverts et de recueillir des informations sur les services s'exécutant sur ces ports.

Hydra :



Un outil d'exploitation couramment utilisé dans le domaine de la sécurité informatique. Il s'agit d'un puissant outil de cracking de mots de passe et de test de force brute. Hydra peut être utilisé pour tester la sécurité des systèmes en tentant de deviner les mots de passe en utilisant différentes combinaisons de caractères.

VirusTotal :



Un service en ligne gratuit qui permet d'analyser des fichiers et des URL pour détecter la présence de malwares. Il utilise une multitude de moteurs antivirus et d'outils d'analyse pour scanner les fichiers et les URL soumis, afin de fournir des rapports détaillés sur les éventuelles menaces détectées.

1.4. Wazuh : Solution de sécurité renommée et utilisée par des utilisateurs de premier plan.

Les grands utilisateurs de la solution Wazuh :

Wazuh est utilisé par de nombreuses organisations à travers le monde, voici quelques exemples de sociétés et d'organisations qui ont adopté Wazuh :

1. **Airbnb** : La plateforme de réservation en ligne Airbnb utilise Wazuh pour renforcer sa sécurité et détecter les activités suspectes sur son infrastructure.
2. **Deloitte** : Le cabinet de conseil et d'audit Deloitte a intégré Wazuh dans ses services de sécurité gérés pour offrir à ses clients une surveillance et une protection avancées.
3. **Verizon Media** : Verizon Media, la société derrière des marques telles que Yahoo et AOL, utilise Wazuh pour la collecte et l'analyse des logs de sécurité, ainsi que pour la détection des menaces.
4. **CERN** : L'Organisation européenne pour la recherche nucléaire (CERN) utilise Wazuh pour la surveillance de sa plateforme informatique et la détection des incidents de sécurité.
5. **Santander Bank** : Santander Bank, une des plus grandes banques en Europe, a déployé Wazuh pour renforcer sa sécurité informatique et détecter les comportements malveillants.

Ces exemples illustrent l'utilisation de Wazuh par certaines grandes sociétés, mais de nombreuses autres entreprises et organisations du monde entier font également confiance à cette solution pour améliorer leur posture de sécurité.

1.5. Les composants de Wazuh

a. Composants globaux :

Le schéma ci-dessous représente les composants Wazuh et le flux de données.

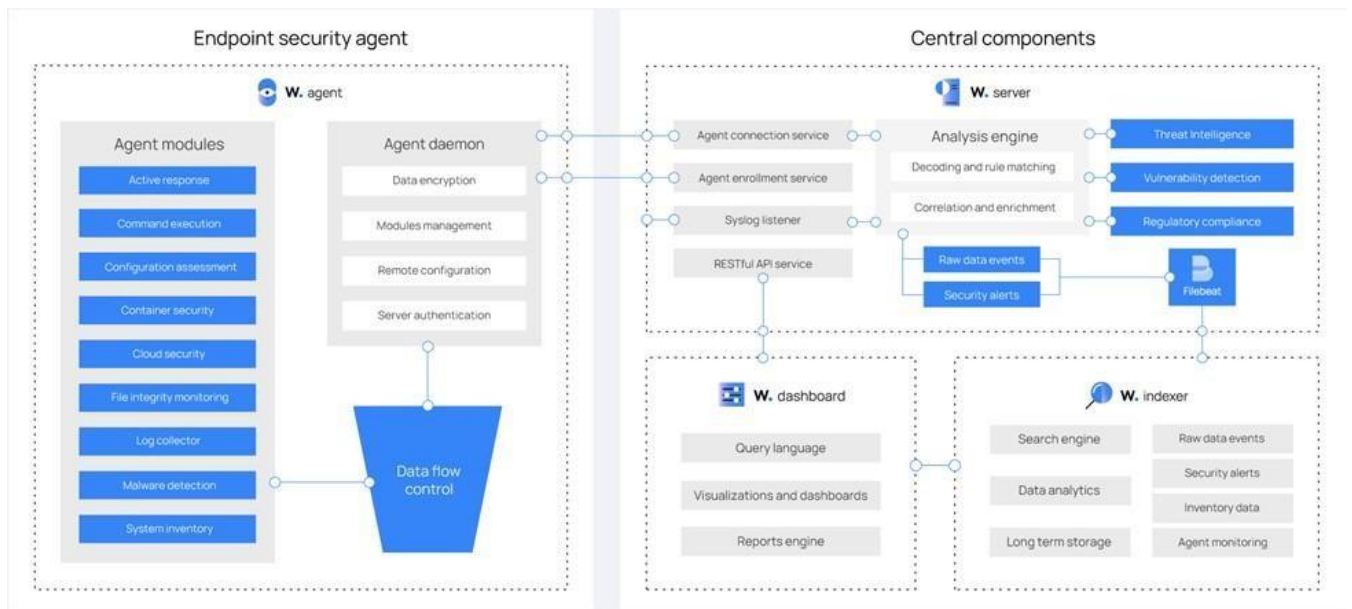


Figure 1 : Composants de Wazuh

b. WAZUH Indexer :

L'indexeur Wazuh est un moteur de recherche et d'analyse en texte intégral hautement évolutif. Ce composant central Wazuh indexe et stocke les alertes générées par le serveur Wazuh et fournit des capacités de recherche et d'analyse de données en temps quasi réel. L'indexeur Wazuh peut être configuré comme un cluster à un ou plusieurs nœuds, offrant évolutivité et haute disponibilité.

L'indexeur Wazuh stocke les données sous forme de documents JSON. Chaque document met en corrélation un ensemble de clés, de noms de champs ou de propriétés, avec leurs valeurs correspondantes qui peuvent être des chaînes, des nombres, des booléens, des dates, des tableaux de valeurs, des géolocalisations ou d'autres types de données.

L'indexeur Wazuh a 4 Types:

- **Wazuh-Alert :** Ceux-ci sont créés chaque fois qu'un événement déclenche une règle avec une priorité suffisamment élevée "le seuil est configurable".
- **wazuh-archives :** Stocke tous les événements (données d'archive) reçus par le serveur Wazuh, qu'ils déclenchent ou non une règle.
- **wazuh-monitoring :** Stocke les données relatives à l'état de l'agent Wazuh au fil du temps et utilisées par l'interface Web pour indiquer quand les agents individuels sont ou ont été **Active**, **Disconnected** ou **Never connected**.
- **wazuh-statistics :** Stocke les données relatives aux performances du serveur Wazuh. Il est utilisé par l'interface web pour représenter les statistiques de performances.

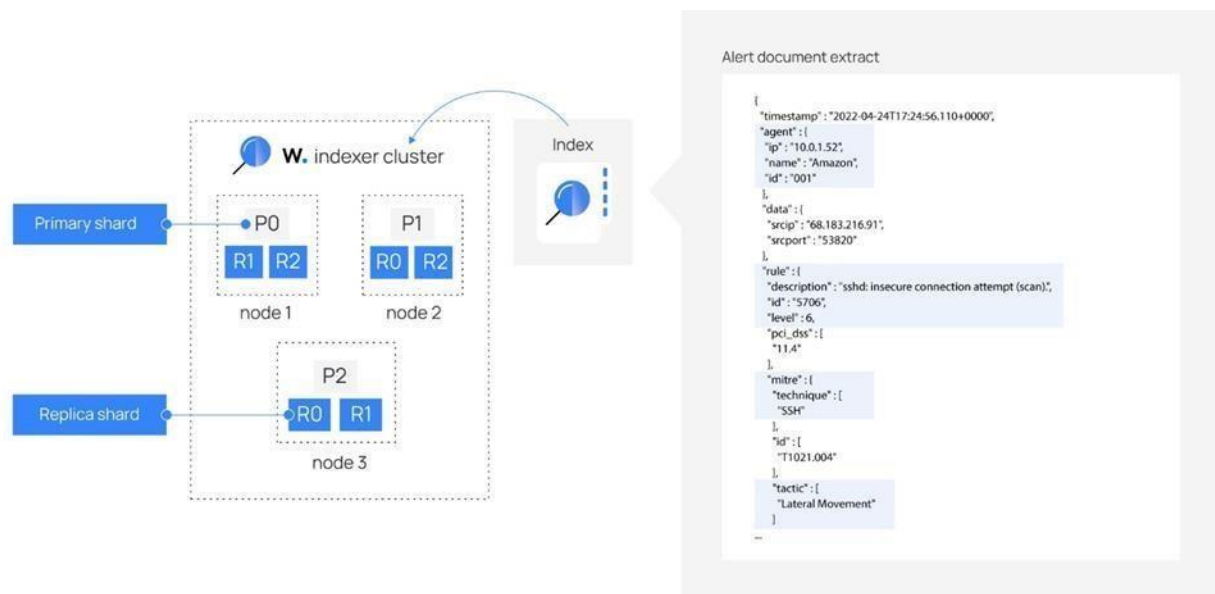


Figure 2 : Architecture de l'indexeur

c. Serveur Wazuh :

- Le composant serveur Wazuh est une plateforme utilisée pour l'analyse des données provenant des agents de sécurité. Son rôle principal est de détecter les menaces et les anomalies, déclenchant des alertes en conséquence. En plus de cela, le serveur Wazuh permet la gestion à distance de la configuration des agents et surveille leur état.
- Pour améliorer ses capacités de détection, le serveur Wazuh utilise des sources de renseignements sur les menaces. Il enrichit également les données d'alerte en utilisant le cadre **MITRE ATT&CK** et les exigences de conformité réglementaire telles que **PCI DSS**, **GDPR**, **HIPAA**, **CIS** et **NIST 800-53**. Cela fournit un contexte précieux pour l'analyse de la sécurité.
- En outre, le serveur Wazuh peut être intégré à des logiciels externes tels que **ServiceNow**, **Jira** et **PagerDuty**, qui sont des systèmes de billetterie utilisés pour la gestion des incidents de sécurité. Il peut également être intégré à des plateformes de messagerie instantanée comme Slack. Ces intégrations permettent de rationaliser les opérations de sécurité et facilitent la communication entre les équipes chargées de la sécurité.
- En résumé, le composant serveur Wazuh est une plateforme qui analyse les données des agents de sécurité, détecte les menaces et les anomalies, et déclenche des alertes. Il utilise des sources de renseignements sur les menaces, enrichit les données d'alerte avec des informations contextuelles et peut être intégré à d'autres logiciels et plateformes pour améliorer les opérations de sécurité.

Architecture du Serveur :

Le serveur Wazuh exécute le moteur d'analyse, l'API RESTful Wazuh, le service d'inscription d'agent, le service de connexion d'agent, le démon de cluster Wazuh et Filebeat. Le

serveur est installé sur un système d'exploitation Linux et s'exécute généralement sur une machine physique autonome, une machine virtuelle, un conteneur Docker ou une instance cloud.

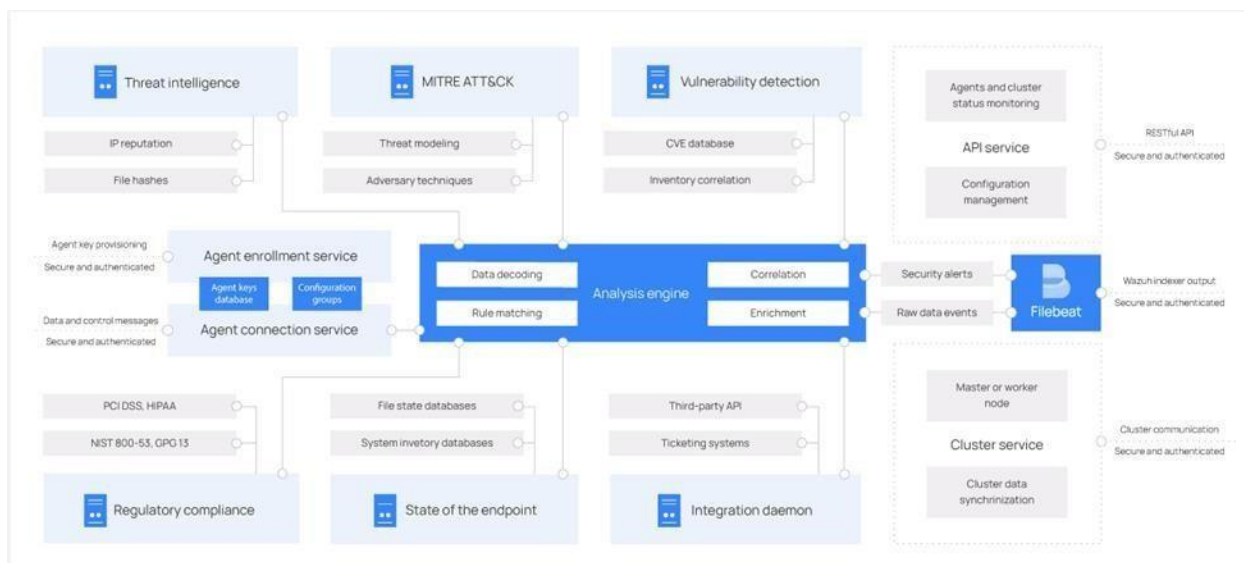


Figure 3 : Architecture du serveur

Les Composants Du serveur :

Le serveur Wazuh comprend plusieurs composants répertoriés ci-dessous qui ont des fonctions différentes, telles que l'inscription de nouveaux agents, la validation de l'identité de chaque agent et le chiffrement des communications entre l'agent Wazuh et le serveur Wazuh.

- **Service d'inscription des agents** : Il est utilisé pour inscrire de nouveaux agents. Ce service fournit et distribue des clés d'authentification uniques à chaque agent. Le processus s'exécute en tant que service réseau et prend en charge l'authentification via des certificats TLS/SSL ou en fournissant un mot de passe fixe.
- **Service de connexion des agents** : Ce service reçoit les données des agents. Il utilise les clés partagées par le service d'inscription pour valider l'identité de chaque agent et chiffrer les communications entre l'agent Wazuh et le serveur Wazuh. De plus, ce service permet la gestion centralisée de la configuration, vous permettant de pousser de nouveaux paramètres d'agent à distance.
- **Moteur d'analyse** : Il s'agit du composant serveur qui effectue l'analyse des données. Il utilise des décodeurs pour identifier le type d'informations traitées (événements Windows, journaux SSH, journaux de serveur Web, et autres). Ces décodeurs extraient également des éléments de données pertinents des messages de journal, tels que l'adresse IP source, l'ID d'événement ou le nom d'utilisateur. Ensuite, en utilisant des règles, le moteur identifie des modèles spécifiques dans les événements décodés qui pourraient déclencher des alertes et éventuellement appeler des contre-mesures automatisées (par exemple, bloquer une adresse IP,

arrêter un processus en cours d'exécution ou supprimer un artefact de logiciel malveillant).

- **API RESTful Wazuh** : Ce service fournit une interface pour interagir avec l'infrastructure Wazuh. Il est utilisé pour gérer les paramètres de configuration des agents et des serveurs, surveiller l'état et la santé globale de l'infrastructure, gérer et éditer les décodeurs et les règles Wazuh, et interroger l'état des points de terminaison surveillés. Le tableau de bord Wazuh l'utilise également.
- **Daemon de cluster Wazuh** : Ce service est utilisé pour mettre à l'échelle les serveurs Wazuh horizontalement en les déployant en tant que cluster. Ce type de configuration, combiné à un répartiteur de charge réseau, assure une haute disponibilité et une répartition de charge. Le daemon de cluster Wazuh est ce que les serveurs Wazuh utilisent pour communiquer entre eux et rester synchronisés.
- **Filebeat** : Il est utilisé pour envoyer des événements et des alertes à l'indexeur Wazuh. Il lit la sortie du moteur d'analyse Wazuh et envoie les événements en temps réel. Il fournit également un équilibrage de charge lorsqu'il est connecté à un cluster d'indexeurs Wazuh à plusieurs nœuds.

d. Wazuh Dashboard :

Wazuh Dashboard : Le tableau de bord Wazuh est une interface utilisateur web flexible et intuitive permettant d'explorer, d'analyser et de visualiser les données des événements de sécurité et des alertes. Il est également utilisé pour la gestion et la surveillance de la plateforme Wazuh. De plus, il propose des fonctionnalités de contrôle d'accès basé sur les rôles (RBAC) et d'authentification unique (SSO).

Visualisation et analyse des données : L'interface web aide les utilisateurs à naviguer à travers les différents types de données collectées par l'agent Wazuh, ainsi que les alertes de sécurité générées par le serveur Wazuh. Les utilisateurs peuvent également générer des rapports et créer des visualisations et des tableaux de bord personnalisés.

Agents monitoring and configuration :

- Le tableau de bord Wazuh offre aux utilisateurs la possibilité de gérer la configuration des agents et de surveiller leur état. Ils peuvent personnaliser les paramètres pour chaque point de terminaison surveillé, tels que les modules d'agent activés, les fichiers journaux à lire, les fichiers à surveiller pour les modifications d'intégrité, ainsi que les vérifications de configuration à effectuer. Cela permet aux utilisateurs d'adapter la surveillance et les fonctionnalités de sécurité en fonction des besoins spécifiques de chaque système ou réseau.

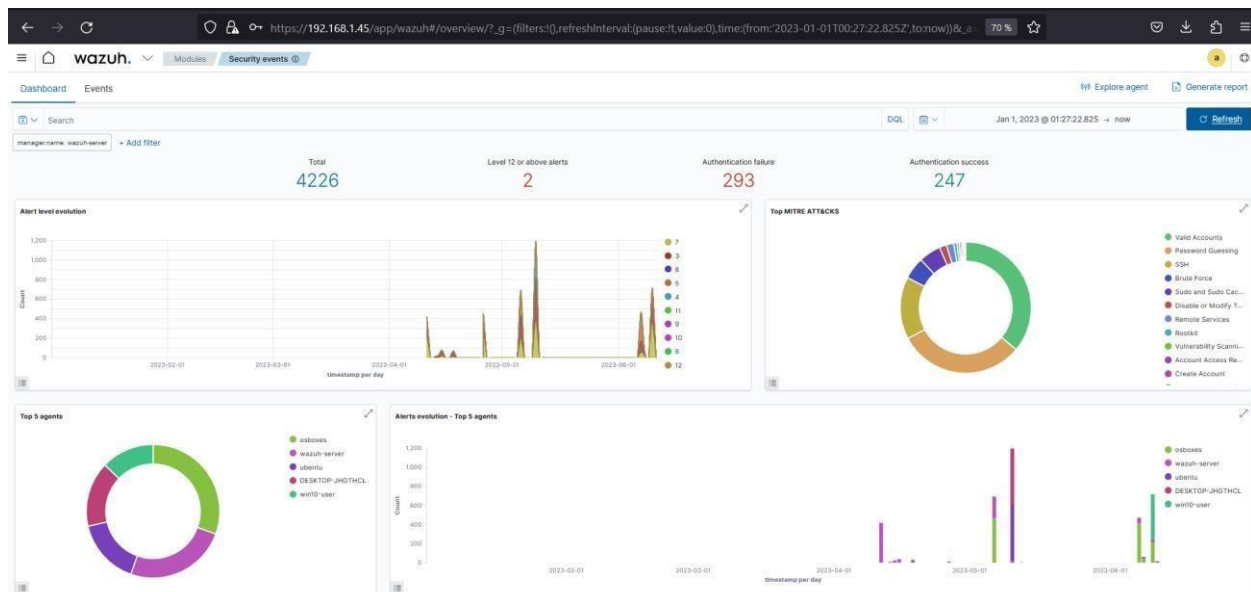


Figure 4: Dashboard Wazuh

Platform management:

- Tableau de bord Wazuh fournit une interface conviviale pour gérer votre déploiement Wazuh. Il permet de surveiller l'état, les journaux et les statistiques des composants de Wazuh, de configurer le serveur Wazuh et de créer des règles et des décodeurs personnalisés pour l'analyse des journaux et la détection des menaces. Cela facilite la gestion et la personnalisation de votre environnement de sécurité.

Developer tool:

- Le tableau de bord Wazuh propose des outils de développement, dont un outil de test des ensembles de règles. Cet outil permet de traiter les messages de journal pour vérifier comment ils sont décodés et s'ils correspondent ou non à une règle de détection de menace. Il est utile pour tester les décodeurs et les règles personnalisés créés par l'utilisateur.

e. Client Wazuh :

L'agent Wazuh s'exécute sur Linux, Windows, macOS, Solaris, AIX et d'autres systèmes d'exploitation. Il peut être déployé sur des ordinateurs portables, des ordinateurs de bureau, des serveurs, des instances cloud, des conteneurs ou des machines virtuelles. L'agent aide à protéger votre système en offrant des capacités de prévention, de détection et de réponse aux menaces. Il est également utilisé pour collecter différents types de données système et applicatives qu'il transmet au serveur Wazuh via un canal crypté et authentifié.

Architecture client :

- L'agent Wazuh a une architecture modulaire. Chaque composant est responsable de ses propres tâches, notamment la surveillance du système de fichiers, la lecture des messages du journal, la collecte des données d'inventaire, l'analyse de la configuration du système et la recherche de logiciels malveillants. Les utilisateurs peuvent gérer les modules d'agent via les paramètres de configuration, en adaptant la solution à leurs cas d'utilisation particuliers.

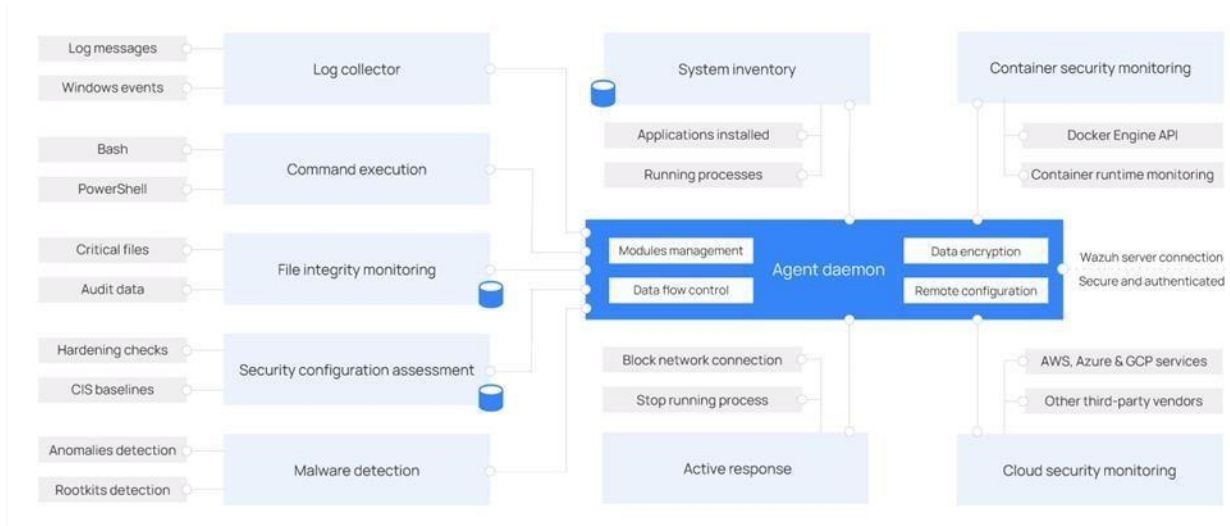


Figure 5 : Architecture du client

Modules du client :

- **Chaque module est responsable de sa propre tâche.**
- **Collecteur de journaux :** Ce composant de l'agent peut lire des fichiers journaux plats et des événements Windows. Il collecte les messages de journal du système d'exploitation et des applications.
- **Exécution de commandes :** Les agents exécutent périodiquement des **commandes** autorisées, collectent leur sortie et la rapportent au serveur Wazuh pour une analyse ultérieure.
- **Surveillance de l'intégrité des fichiers (FIM) :** Ce module surveille le système de fichiers et signale la création, la suppression ou la modification de fichiers. Il suit les modifications des attributs, des autorisations, de la propriété et du contenu des fichiers. Lorsqu'un événement se produit, il capture en temps réel les détails de qui, quoi et quand.
- **Évaluation de la configuration de sécurité (SCA) :** Ce composant fournit une évaluation continue de la configuration en utilisant des vérifications prédéfinies basées sur les normes du Center of Internet Security (CIS).
- **Inventaire système :** Ce module de l'agent effectue périodiquement des analyses, collectant des données d'inventaire telles que la version du système d'exploitation, les interfaces réseau, les processus en cours d'exécution, les applications installées et une liste des ports ouverts. Les analyses sont stockées dans des bases de données sqlite.
- **Détection de logiciels malveillants :** En utilisant une approche non basée sur les signatures, ce composant est capable de détecter les anomalies et la possible

présence de rootkits. Il recherche également les processus cachés, les fichiers cachés et les ports cachés tout en surveillant les appels système.

- **Réponse active** : Ce module exécute des actions automatiques lorsque des menaces sont détectées, déclenchant des réponses pour bloquer une connexion réseau, arrêter un processus en cours, ou supprimer un fichier malveillant. Vous pouvez utiliser vos propres réponses.

Chapitre 2: Environnement Wazuh

2.1. Introduction

Ce chapitre présente la configuration de l'environnement Wazuh, incluant l'installation des serveurs et clients. Les exigences matérielles et logicielles sont abordées, ainsi que l'architecture système recommandée. Nous détaillons ensuite l'installation du serveur Wazuh et des clients sur Linux et Windows, en mettant l'accent sur les différences de configuration. Enfin, nous testons les règles de détection par défaut pour garantir leur efficacité dans la détection des menaces potentielles.

2.2. Environnement de déploiement

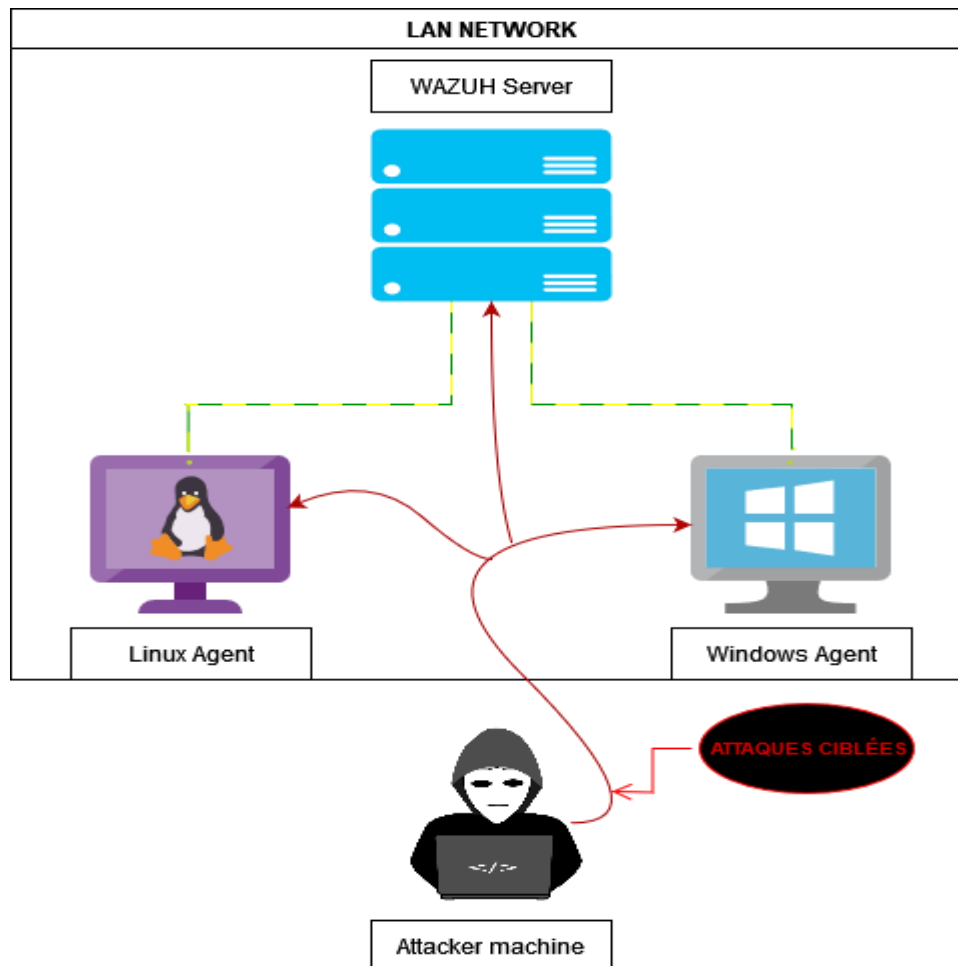


Figure 6 : Schéma de déploiement

L'architecture comprend un serveur Wazuh central qui collecte, stocke et analyse les données de sécurité provenant de deux agents, l'un pour Linux et l'autre pour Windows. Les agents

surveillent les journaux système, les événements de sécurité et d'autres informations pertinentes, et envoient ces données au serveur Wazuh. Le serveur Wazuh utilise des règles de détection pour identifier les activités suspectes ou malveillantes. Si une activité suspecte est détectée, le serveur Wazuh génère des alertes pour informer les administrateurs de la sécurité. Dans cette architecture, un attaquant externe tente de mener des attaques contre le système surveillé, utilisant des techniques telles que l'injection SQL ou les attaques par force brute. L'objectif de l'architecture est de renforcer la sécurité du système surveillé en détectant les attaques et en prenant les mesures appropriées pour y répondre.

1. **Serveur Wazuh** : C'est le cœur de l'architecture. Le serveur Wazuh est responsable de la collecte, du stockage et de l'analyse des données de sécurité provenant des agents. Il fournit également des fonctionnalités de détection des menaces, de gestion des incidents et de génération d'alertes. Le serveur Wazuh est configuré avec des règles de détection pour identifier les comportements suspects ou malveillants.
2. **Agent Linux** : Cet agent est installé sur un système Linux qui doit être surveillé. L'agent collecte les journaux système, les événements de sécurité et les autres informations pertinentes et les envoie au serveur Wazuh pour l'analyse. L'agent Linux est configuré pour surveiller les activités du système et détecter les anomalies ou les signes d'attaques.
3. **Agent Windows** : Cet agent est installé sur un système Windows qui doit être surveillé. Il collecte les journaux d'événements, les fichiers de registre et d'autres données de sécurité importantes et les envoie au serveur Wazuh pour l'analyse. L'agent Windows est également configuré pour détecter les comportements suspects ou malveillants et générer des alertes en conséquence.
4. **Attaquant** : Il s'agit d'une entité externe qui tente de mener des attaques contre le système surveillé. L'attaquant peut utiliser différentes techniques d'attaque telles que l'injection SQL, les attaques par force brute, les scans de ports, etc. L'objectif de l'attaquant est de compromettre le système ou de causer des dommages.

2.3. Installation du serveur Wazuh

La machine virtuelle Wazuh (OVA) a les spécifications et les exigences suivantes : a.

Spécifications :

- Distribution: CentOS 7
- Architecture: 64 bits
- Format VM: OVA
- Version: 4.4.3
- Composants: Wazuh manager 4.4.3, Wazuh indexer 4.4.3, Filebeat-OSS 7.10.2, Wazuh dashboard 4.4.

b. Exigences matérielles :

1. Le système d'exploitation hôte doit être un système 64 bits.
2. La virtualisation matérielle doit être activée dans le micrologiciel de l'hôte.
3. Une plateforme de virtualisation, telle que VirtualBox, doit être installée sur le système hôte.

c. La machine virtuelle Wazuh est préconfigurée avec les spécifications suivantes :

CPU (coeurs) : 4
RAM (Go) : 8
Stockage (Go) : 50

d. Pour importer et accéder à la machine virtuelle Wazuh :

1. Téléchargez le fichier OVA de Wazuh depuis le site Web de Wazuh.
2. Importez le fichier OVA dans votre plateforme de virtualisation, comme VirtualBox.
3. Démarrez la machine virtuelle importée.

Une fois la machine virtuelle démarrée, vous pouvez accéder à celle-ci et au tableau de bord de Wazuh en suivant ces étapes :

1. Utilisez les informations d'identification suivantes pour accéder à la machine virtuelle :
 - Utilisateur : **wazuh-user**
 - Mot de passe : **wazuh**
2. Pour accéder au tableau de bord de Wazuh via l'interface Web, utilisez les informations d'identification suivantes :
 - URL: **https://<adresse_ip_du_serveur_wazuh>**
 - Utilisateur: **admin**
 - Mot de passe: **admin**

e. Concernant les fichiers de configuration, voici leurs emplacements :

- Wazuh manager : `/var/ossec/etc/ossec.conf`
- Wazuh indexer : `/etc/wazuhindexer/opensearch.yml` • Filebeat-OSS :
`/etc/filebeat/filebeat.yml` • Tableau de bord de Wazuh :
 - `/etc/wazuh-dashboard/opensearch_dashboards.yml`
 - `/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml`

Veuillez noter que vous pouvez personnaliser ces fichiers de configuration selon vos besoins.

2.4. Installation des clients Wazuh

a. LINUX Client :

Ajoutez le dépôt Wazuh :

1. Installer la clé GPG :

Dans le contexte de Wazuh, GPG (GNU Privacy Guard) est utilisé pour la sécurisation des communications entre les différents composants de la solution. Wazuh utilise des clés GPG pour chiffrer les données sensibles telles que les journaux (logs) collectés, afin de garantir leur confidentialité et leur intégrité.

```

root@osboxes:/home/osboxes# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring
gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: no valid OpenPGP data found.
gpg: Total number processed: 0

```

Figure 7 : Installation de la clé GPG

2. Ajouter le dépôt :

Dans le cadre de l'installation et de la configuration de Wazuh, il est nécessaire d'ajouter le dépôt (repository) correspondant à votre système d'exploitation. Le dépôt contient les packages et les mises à jour nécessaires pour installer et maintenir Wazuh à jour.

```

root@osboxes:/home/osboxes# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main

```

Figure 8 : Ajout du dépôt

3. Mettre à jour les informations du package :

Pour mettre à jour les informations du package dans Wazuh, vous pouvez utiliser la commande appropriée en fonction de votre système d'exploitation.

```

root@osboxes:/home/osboxes# apt-get update
Get:1 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:2 http://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Hit:3 http://deb.debian.org/debian bullseye InRelease
Get:4 https://packages.wazuh.com/4.x/apt stable InRelease [17.3 kB]
Get:5 http://security.debian.org/debian-security bullseye-security/main Sources [201 kB]
Get:6 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [245 kB]
Get:7 http://security.debian.org/debian-security bullseye-security/main Translation-en [161 kB]
Get:8 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [29.0 kB]
Fetched 745 kB in 1s (964 kB/s)
Reading package lists... Done
root@osboxes:/home/osboxes#

```

Figure 9 : Mis à jour des packages

Déployer un client Wazuh :

1. Installer le client

Pour déployer l'agent Wazuh sur votre point de terminaison, sélectionnez votre gestionnaire de packages et modifiez la variable **WAZUH_MANAGER** pour qu'elle contienne l'adresse IP ou le nom d'hôte de votre Serveur Wazuh :


```

root@osboxes:/home/osboxes# WAZUH_MANAGER="192.168.1.45" apt-get install wazuh-agent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  wazuh-agent
1 upgraded, 0 newly installed, 0 to remove and 115 not upgraded.
Need to get 8,905 kB of archives.
After this operation, 223 kB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-agent amd64 4.4.3-1 [8,905 kB]
Fetched 8,905 kB in 1s (7,236 kB/s)
Reading changelogs... Done
Preconfiguring packages ...
(Reading database ... 39767 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.4.3-1_amd64.deb ...
Unpacking wazuh-agent (4.4.3-1) over (4.4.1-1) ...
Setting up wazuh-agent (4.4.3-1) ...
root@osboxes:/home/osboxes# █

```

Figure 10 : Installation du client "Linux"

2. Activez et démarrez le service d'agent Wazuh :

```

root@osboxes:/home/osboxes# systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
Synchronizing state of wazuh-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-agent
root@osboxes:/home/osboxes# █

```

Figure 11 : Activation et Démarrage du service agent

3. Vérifier le statut de votre agent :

```

root@osboxes:/home/osboxes# systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-06-10 12:35:17 EDT; 5min ago
     Tasks: 30 (limit: 2341)
    Memory: 17.6M
       CPU: 13.114s
    CGroup: /system.slice/wazuh-agent.service
            └─3602 /var/ossec/bin/wazuh-execd
               3613 /var/ossec/bin/wazuh-agentd
               3627 /var/ossec/bin/wazuh-syscheckd
               3639 /var/ossec/bin/wazuh-logcollector
               3656 /var/ossec/bin/wazuh-modulesd

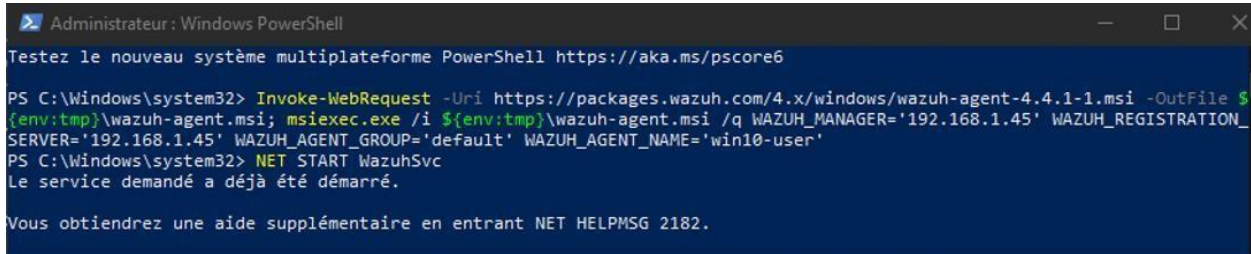
Jun 10 12:35:09 osboxes systemd[1]: Starting Wazuh agent...
Jun 10 12:35:09 osboxes env[3579]: Starting Wazuh v4.4.3...
Jun 10 12:35:10 osboxes env[3579]: Started wazuh-execd...
Jun 10 12:35:11 osboxes env[3579]: Started wazuh-agentd...
Jun 10 12:35:13 osboxes env[3579]: Started wazuh-syscheckd...
Jun 10 12:35:14 osboxes env[3579]: Started wazuh-logcollector...
Jun 10 12:35:15 osboxes env[3579]: Started wazuh-modulesd...
Jun 10 12:35:17 osboxes env[3579]: Completed.
Jun 10 12:35:17 osboxes systemd[1]: Started Wazuh agent.
root@osboxes:/home/osboxes# █

```

Figure 12 : Vérification du status Wazuh

b. WINDOWS Client :

Pour déployer l'agent Wazuh sur votre point de terminaison, choisissez l'une des alternatives du shell de commande et modifiez la variable WAZUH_MANAGER afin qu'elle contienne l'adresse IP ou le nom d'hôte du gestionnaire Wazuh.



```
Administrateur : Windows PowerShell
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.4.1-1.msi -OutFile $(env:tmp)\wazuh-agent.msi; msisexec.exe /i $(env:tmp)\wazuh-agent.msi /q WAZUH_MANAGER='192.168.1.45' WAZUH_REGISTRATION_SERVER='192.168.1.45' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='win10-user'
PS C:\Windows\system32> NET START WazuhSvc
Le service demandé a déjà été démarré.

Vous obtiendrez une aide supplémentaire en entrant NET HELPMSG 2182.
```

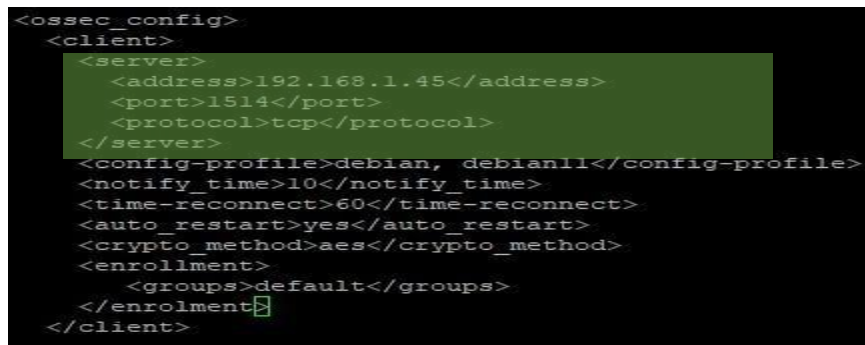
Figure 13 : Déploiement du client "Windows"

2.5. Connectivité des clients avec le serveur

a. Configuration du client:

Lancez le terminal en tant qu'utilisateur root, modifiez le fichier de configuration de l'agent `/var/ossec/etc/ossec.conf` et apportez les modifications suivantes :

- Inclure l'adresse IP ou le nom DNS du gestionnaire Wazuh dans la section suivante :



```
<ossec_config>
  <client>
    <server>
      <address>192.168.1.45</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>debian, debianll</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
    <enrollment>
      <groups>default</groups>
    </enrollment>
  </client>
```

Figure 14: Configuration du client

b. Résultats de connectivité sur dashboard:

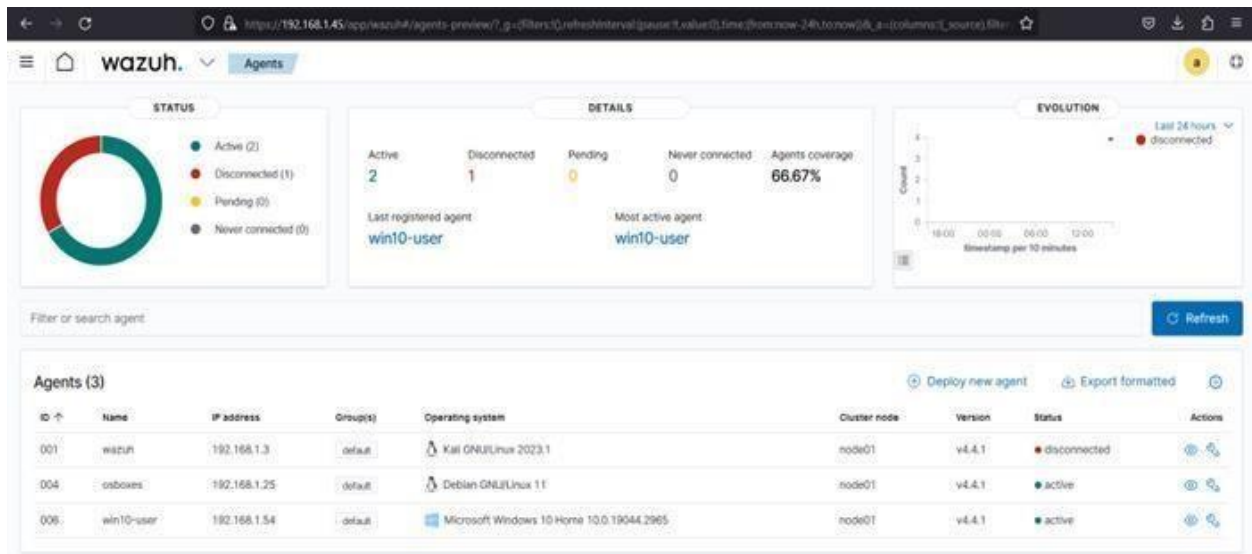


Figure 15 : Résultat de connectivité

2.6. Test des règles par défaut

Lors du déploiement de Wazuh, il est essentiel de vérifier les règles par défaut pour garantir la conformité et la sécurité de votre environnement. Ce test consiste à évaluer l'efficacité des règles par défaut en détectant les menaces et les comportements suspects. Il permet de s'assurer que Wazuh est correctement configuré pour surveiller et signaler les activités potentiellement malveillantes. En effectuant ce test, vous pouvez identifier les éventuelles lacunes dans les règles par défaut et prendre les mesures nécessaires pour les combler, renforçant ainsi la sécurité de votre système.

Quelque exemple détecter avec Wazuh :

SSH : La détection des intrusions SSH sur Wazuh est une fonctionnalité qui surveille et identifie les activités suspectes liées aux connexions SSH. Elle vise spécifiquement à repérer les tentatives non autorisées d'accès à un système à distance via le protocole SSH.

Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 7, 2023 @ 15:30:50.501	000	wazuh-server	T1021.004	Lateral Movement	sshd: insecure connection attempt (scan).	6	5706

Table

JSON

Rule

@timestamp	2023-06-07T14:30:50.501Z
_id	i8ZElogB-c-5nNcSnCFS
agent.id	000
agent.name	wazuh-server
data.srcip	192.168.11.241
data.srcport	63172
decoder.name	sshd
decoder.parent	sshd
full_log	Jun 7 14:30:49 wazuh-server sshd[10179]: Did not receive identification string from 192.168.11.241 port 63172

Figure 16 : Détection des connexions SSH

Privilege Escalation : La détection de l'escalade de privilèges sur Wazuh est un mécanisme qui permet de surveiller les comportements suspects liés à l'obtention de privilèges d'accès plus élevés sur un système.

Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 7, 2023 @ 14:26:50.583	000	wazuh-server	T1110.001 T1021.004 T1078	Credential Access, Lateral Movement, Defense Evasion, Persistence, Privilege Escalation, Initial Access	sshd: Attempt to login using a non-existent user	5	5710
<div>TableJSONRule</div>							
@timestamp	2023-06-07T13:26:50.583Z						
_id	GsYJlogB-c-5nNcS7yEj						
agent.id	000						
agent.name	wazuh-server						
data.srcip	192.168.11.241						
data.srcuser	wazuh-server						
decoder.name	sshd						
decoder.parent	sshd						

Figure 17 : Détection d'escalade de privilèges

Brute force : La détection des attaques brute-force se concentre sur les tentatives répétées de deviner un mot de passe en essayant différentes combinaisons

May 6, 2023 @ 18:15:49.832	000	wazuh-server	T1110	Credential Access	sshd: brute force trying to get access to the system. Authentication failed.	10	5763
Table	JSON	Rule					
@timestamp	2023-05-06T17:15:49.832Z						
_id	JN4Q8ocB481LizEVH37M						
agent.id	000						
agent.name	wazuh-server						
data.dstuser	root						
data.srcip	192.168.1.8						
data.srport	43302						
decoder.name	sshd						
decoder.parent	sshd						
full_log	May 6 17:15:48 wazuh-server sshd[3437]: Failed password for root from 192.168.1.8 port 43302 ssh2						
id	1683393349.853291						
input.type	log						
location	/var/log/secure						

Figure18 : Détection des bruteforce

Hydra bruteforce : Cette règle est spécifique à l'outil Hydra et détecte les tentatives de force brute effectuées avec Hydra. Elle peut identifier les attaques visant à deviner les mots de passe en utilisant différentes combinaisons de noms d'utilisateur et de mots de passe. On remarque des noms d'utilisateur aléatoires par exemple ' qdjlqldjdq '

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Apr 18, 2023 @ 14:57:07.975	002	salah	T1110.001 T1021.004 T1078	Credential Access, Lateral Movement, Defense Evasion, Persistence, Privilege Escalation, Initial Access	sshd: Attempt to login using a non-existent user	5	5710

Table	JSON	Rule
-------	------	------

@timestamp	2023-04-18T14:57:07.975Z
_id	g63ellcBxpZHjHqgvH
agent.id	002
agent.ip	172.16.5.2
agent.name	salah
data.srcip	172.16.5.2
data.srcuser	qdjlqldjdq
decoder.name	sshd
decoder.parent	sshd
full_log	Apr 18 10:57:09 salah sshd[24204]: Failed password for invalid user qdjlqldjdq from 172.16.5.2 port 56372 ssh2
id	1681829827.57089

Figure19 : Détection de Hydra bruteforce

NMAP SCAN : Wazuh propose des règles de détection pour identifier les scans de port réalisés à l'aide de Nmap. Ces règles permettent de repérer les tentatives d'exploration de réseau et d'identification des services disponibles sur les hôtes cibles.

Lorsqu'un scan Nmap est détecté, Wazuh génère une alerte qui inclut des informations telles que l'adresse IP source, le port cible, le protocole utilisé et d'autres détails pertinents. Cette alerte permet aux administrateurs de sécurité de prendre des mesures appropriées pour enquêter sur l'activité suspecte et éventuellement bloquer les adresses IP sources.

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 7, 2023 @ 18:49:18.583	004	osboxes	T1595.002	Reconnaissance	Multiple web server 400 error codes from same source ip.	10	31151
<div> <div>Table</div> <div>JSON</div> <div>Rule</div> </div>							
@timestamp	2023-06-07T17:49:18.583Z						
_id	DBb8logB-c-5nNcSWiMq						
agent.id	004						
agent.ip	192.168.137.160						
agent.name	osboxes						
data.id	405						
data.protocol	PROPFIND						
data.srcip	192.168.137.222						
data.url	/						
decoder.name	web-accesslog						
full_log	192.168.137.222 - - [07/Jun/2023:13:49:17 -0400] "PROPFIND / HTTP/1.1" 405 528 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"						

Figure 20 : Détection de scan NMAP

Rootkit : Les règles de détection au niveau du noyau dans Wazuh surveillent les composants système critiques et les comportements susceptibles d'indiquer la présence d'un rootkit. Il s'agit notamment de modifications suspectes des modules du noyau, de processus masqués, de modifications non autorisées des appels système, d'activités réseau anormales et d'autres indicateurs compromettants au niveau du noyau.

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Apr 18, 2023 @ 14:55:57.660	002	salah	T1014	Defense Evasion	Possible kernel level rootkit.	11	521
<div> <div>Table</div> <div>JSON</div> <div>Rule</div> </div>							
@timestamp	2023-04-18T14:55:57.660Z						
_id	dq3dlcBxpZHjHqjQuN						
agent.id	002						
agent.ip	172.16.5.2						
agent.name	salah						
data.title	Anomaly detected in file '/root/.cache/doc'.						
decoder.name	rootcheck						
full_log	Anomaly detected in file '/root/.cache/doc'. Hidden from stats, but showing up on readdir. Possible kernel level rootkit.						
id	1681829757.50949						
input.type	log						
location	rootcheck						

Figure 21 : Détection des Rootkits

Installation d'un nouveau fichier : Lorsqu'un nouveau fichier est installé sur un système, Wazuh peut analyser différents aspects de cette action, tels que l'emplacement du fichier, les permissions accordées, l'origine du fichier, les actions associées (par exemple, la création d'un nouveau service ou l'ajout d'une tâche planifiée), et d'autres informations pertinentes.

Jun 8, 2023 @ 17:30:20.359	004	osboxes	New dpkg (Debian Package) installed.	7	2902																																													
<table><tr><td>Table</td><td>JSON</td><td>Rule</td></tr><tr><td>@timestamp</td><td colspan="5">2023-06-08T16:30:20.359Z</td></tr><tr><td>_id</td><td colspan="5">wUrYm4gBgqr3kHZeSi04</td></tr><tr><td>agent.id</td><td colspan="5">004</td></tr><tr><td>agent.ip</td><td colspan="5">192.168.11.149</td></tr><tr><td>agent.name</td><td colspan="5">osboxes</td></tr><tr><td>data.arch</td><td colspan="5">amd64</td></tr><tr><td>data.dpkg_status</td><td colspan="5">status installed</td></tr></table>						Table	JSON	Rule	@timestamp	2023-06-08T16:30:20.359Z					_id	wUrYm4gBgqr3kHZeSi04					agent.id	004					agent.ip	192.168.11.149					agent.name	osboxes					data.arch	amd64					data.dpkg_status	status installed				
Table	JSON	Rule																																																
@timestamp	2023-06-08T16:30:20.359Z																																																	
_id	wUrYm4gBgqr3kHZeSi04																																																	
agent.id	004																																																	
agent.ip	192.168.11.149																																																	
agent.name	osboxes																																																	
data.arch	amd64																																																	
data.dpkg_status	status installed																																																	

Figure 22 : Détection De l'installation d'un nouveau package

Détection de vulnérabilités : Cet événement indique qu'un fichier spécifique, '/root/bash.sh', est détenu par l'utilisateur root et possède des permissions d'écriture accordées à tous les utilisateurs. Cette situation peut être potentiellement dangereuse, car cela signifie que n'importe quel utilisateur peut modifier ce fichier, ce qui peut représenter un risque pour la sécurité du système.

Apr 18, 2023 @ 14:55:57.808	002	salah	Host-based anomaly detection event (rootcheck).	7	510
Table	JSON	Rule			
@timestamp	2023-04-18T14:55:57.808Z				
_id	eK3dlcBxpZHqJHqQuN				
agent.id	002				
agent.ip	172.16.5.2				
agent.name	salah				
data.file	/root/bash.sh				
data.title	File is owned by root and has written permissions to anyone.				
decoder.name	rootcheck				
full_log	File '/root/bash.sh' is owned by root and has written permissions to anyone.				

Figure 23: Détection de vulnérabilités

Chapitre 3 : Renforcement de la sécurité avec Wazuh

3.1 Installation des clients Wazuh :

L'ajout de règles de détection supplémentaires permet d'améliorer la capacité d'un système de détection à détecter une plus large gamme de menaces et d'activités malveillantes. Ces règles peuvent être personnalisées en fonction des besoins spécifiques de l'organisation et des types de menaces auxquelles elle est confrontée.

- **Les règles** sont utilisées pour détecter des schémas spécifiques d'événements ou de comportements malveillants. Elles permettent d'identifier les activités suspectes ou les violations de sécurité et de générer des alertes en conséquence.
- **Les décodeurs** sont responsables du traitement initial des événements bruts collectés à partir des journaux système ou des flux de données. Ils analysent les données brutes et les transforment en un format compréhensible par le système de détection d'incidents.

Voici quelques exemples de règles de détection supplémentaires qui peuvent être ajoutées :

a. Tentative infructueuse de négociation SSH :

Description : Cette règle indique qu'une tentative de négociation SSH depuis l'adresse IP 192.168.137.222 a échoué. L'échange de bannière a été effectué, mais la version du protocole n'a pas pu être lue, ce qui peut indiquer une tentative d'authentification infructueuse ou une erreur lors de la négociation du protocole SSH.

Cette règle peut être utilisée pour détecter les tentatives d'accès non autorisées via SSH et peut être une indication d'une activité suspecte ou malveillante. Il est recommandé d'analyser les journaux et de prendre les mesures appropriées pour sécuriser le système en cas de telles tentatives

Règle associée :

```
<group name="SSHD">
<!-- 110000 to 110020-->

<rule id="110000" level="5">
  <decoded_as>sshd</decoded_as>
  <description>Unsuccessful SSH negotiation $(srcip)</description>
</rule>

<rule id="110001" level="7">
  <if_sid>110000</if_sid>
  <match>
    <regex>sshd</regex>
    Unable to negotiate with %src_ip% port %port%
  </match>
  <description>Unsuccessful SSH negotiation from %src_ip%</description>
  <group>sshd</group>
</rule>
</group>
```

Figure 24 : Règles de négociation SSH

Ce code représente une partie des règles de détection liées au service SSH (Secure Shell Daemon - démon de shell sécurisé) dans le système Wazuh. Voici une explication de chaque élément :

- `<group name="SSHD">` : Ceci définit un groupe de règles spécifique pour le service SSH.
 - `<rule id="110000" level="5">` : C'est la première règle de détection. Elle a un ID de règle de "110000" et un niveau de sévérité de "5" (échelle de 1 à 10, où 1 est la sévérité la plus faible). Elle est associée au décodage des événements SSH.
- `<decoded_as>sshd</decoded_as>` : Cet élément indique que la règle est décodée en tant qu'événement SSH.
- `<description>Unsuccessful SSH negotiation $(srcip)</description>` : C'est la description de la règle. Elle indique qu'une négociation SSH infructueuse a eu lieu avec l'adresse IP source (indiquée par "\$ (srcip)").
- `<rule id="110001" level="7">` : C'est la deuxième règle de détection, qui dépend de la règle précédente (id="110000"). Elle a un niveau de sévérité de "7".
- `<if_sid>110000</if_sid>` : Cet élément spécifie que cette règle est conditionnelle à l'ID de la règle précédente (id="110000").
- `<match>` : C'est la condition de correspondance de la règle. Elle utilise une expression régulière pour vérifier si le texte "sshd Unable to negotiate with %src_ip% port %port%" est présent dans les événements.
 - `<description>Unsuccessful SSH negotiation from %src_ip%</description>` : C'est la description de la règle. Elle indique qu'une négociation SSH infructueuse a été détectée à partir de l'adresse IP source (%src_ip%).
- `<group>sshd</group>` : Cet élément spécifie que cette règle fait partie du groupe "sshd".

Décodeurs associés :

```
<decoder name="local_decoder_example">
  <program_name>local_decoder_example</program_name>
</decoder>
<decoder name="sshd">
  <parent>sshd</parent>
  <regex offset="afterparent">(Unable to negotiate with)</regex>
  <order>action</order> </decoder> <decoder name="sshd">
  <parent>sshd</parent>
  <regex offset="afterparent">(\w+.\w+.\w+.\w+)</regex>
  <order>srcip</order> </decoder> <decoder name="sshd">
  <parent>sshd</parent>
  <regex offset="afterparent">(\w+.\w+.\w+.\w+)</regex>
  <order>ip_address</order>
</decoder>
```

Figure 25 : Décodeur associé

Afin de repérer les tentatives infructueuses de connexion SSH, nous utilisons un décodeur qui analyse les journaux provenant des clients. Le premier décodeur est spécifiquement conçu pour détecter le motif "Unable to negotiate with", tandis que le deuxième utilise une expression régulière pour extraire l'adresse IP source.

b. Résultat de la règle de négociation SSH :

Jun 7, 2023 @ 18:53:32.842	004	osboxes	Unsuccessful SSH negotiation 192.168.137.222	5	110000
Table	JSON	Rule			
@timestamp	2023-06-07T17:53:32.842Z				
_id	ecb-logB-c-5nNcSLyMI				
agent.id	004				
agent.ip	192.168.137.160				
agent.name	osboxes				
data.ip_address	192.168.137.222				
data.srcip	192.168.137.222				
decoder.name	sshd				
full_log	Jun 7 13:53:32 osboxes sshd[10545]: banner exchange: Connection from 192.168.137.222 port 41194: could not read protocol version				

Figure 26 : Résultat de la règle de négociation SSH

3.2. Active Response

Active response : La fonction de réponse active (active response) dans Wazuh est une capacité qui permet de prendre des mesures immédiates en réponse à une activité malveillante détectée. Plutôt que de simplement signaler les incidents, la réponse active permet à Wazuh d'effectuer des actions automatiques pour contrer les attaques ou réduire leur impact.

Les mesures de réponse active peuvent inclure des actions telles que le blocage d'une adresse IP source suspecte, la désactivation d'un compte utilisateur compromis, la mise en quarantaine d'un système infecté, ou même la déconnexion d'une session SSH malveillante. [Configuration de l'active response](#) :

Vérifiez la configuration du bloc `<command>` dans le fichier de configuration `/var/ossec/etc/ossec.conf` du serveur Wazuh. Ajoutez-en un s'il n'existe pas déjà.

```
<!--
<active-response>
| active-response options here
</active-response>
-->
<active-response>
  <command>firewall-drop</command>
  <location>localhost</location>
  <rules_id>5740</rules_id>
  <rules_id>110000</rules_id>
  <timeout>1000</timeout>
</active-response>
```

Explication de la configuration de l'active response : Pour configurer la réponse active, nous commençons par spécifier la commande que nous souhaitons utiliser, dans notre cas, "firewall-drop". Ensuite, nous déterminons les conditions qui déclenchent cette réponse active, telles que l'activation des règles 5740 et

110000.

Figure 27 : Configuration de l'active response

Avant la configuration de l'active response:

Explication du scan nmap : On a effectué le scan à l'aide de Nmap sur la machine victime, et nous avons obtenu des informations détaillées telles que les ports ouverts, les versions du système d'exploitation, les adresses MAC, etc.

```
(root@salah)~# nmap -sC -sV -T4 192.168.137.160 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-07 13:53 EDT
Nmap scan report for osboxes.mshome.net (192.168.137.160)
Host is up (0.00093s latency).
Not shown: 987 filtered tcp ports (no-response), 12 filtered tcp ports (admin-prohibited)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 8e:4f:77:7f:f6:aa:6a:dc:17:c9:bf:5a:2b:eb:8c:41 (RSA)
|   256 a3:9c:66:73:fc:b9:23:c0:0f:da:1d:c9:84:d6:b1:4a (ECDSA)
|_  256 6d:c2:0e:89:25:55:10:a9:9e:41:6e:0d:81:9a:17:cb (ED25519)
MAC Address: 08:00:27:94:AC:E9 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.79 seconds
```

Figure 28 : Succès du scan NMAP

Après la configuration de l'active response:

L'adresse IP de l'attaquant est maintenant bloquée 'les scans n'affichent pas les données de la machine et les pings vers la machine client ne marchent pas' :

Jun 7, 2023 @ 18:37:46.758	004	osboxes	Host Blocked by firewall-drop Active Response	3	651
Table	JSON	Rule			
@timestamp		2023-06-07T17:37:46.758Z			
_id		3MbviogB-c-5nNcSqSKd			
agent.id		004			
agent.ip		192.168.137.160			
agent.name		osboxes			
data.command		add			
data.origin.module		wazuh-execd			
data.origin.name		node01			
data.parameters.alert.agent.id		004			

Figure 29 : Détection du scan NMAP

Explication : Suite à la mise en place de l'active response, l'adresse IP de l'attaquant est désormais bloquée. En conséquence, les scans ne révèlent plus les informations de la machine et les tentatives de ping vers la machine client échouent.

```
(root@salah)-[~]
# nmap -sC -sV -T4 192.168.137.160 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-07 13:55 EDT
Nmap scan report for osboxes.mshome.net (192.168.137.160)
Host is up (0.00077s latency).
All 1000 scanned ports on osboxes.mshome.net (192.168.137.160) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:94:AC:E9 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.14 seconds
```

Figure 30 : Echec du scan nmap

```
(root@salah)-[~]
# ping 192.168.11.160
PING 192.168.11.160 (192.168.11.160) 56(84) bytes of data.

```

Figure 31 : Echec des pings

3.3. VirusTotal

a. La configuration du VirusTotal :

Coté Client :

1. Ajoutez une entrée dans le bloc <syscheck> pour configurer la surveillance en quasi temps réel d'un répertoire. Dans ce cas, vous surveillez le répertoire **/root**.

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
  <directories realtime="yes">/root</directories>
  <directories check_all="yes" realtime="yes">/root/important_files</directories>
```

Figure 32 : Configuration de la surveillance

2. Installer **jq**, un utilitaire qui traite les entrées **JSON** dans le script de l'active response:

```
sudo apt update
sudo apt -y install jq
```

Figure 33: Installation jq

3. Créez le script de réponse active **/var/ossec/active-response/bin/remove-threat.sh** pour supprimer les fichiers malveillants du point de terminaison :

```
#!/bin/bash
LOCAL='dirname $0';
cd $LOCAL
cd ../
PWD='pwd'

read INPUT_JSON
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.data.virustotal.source.file)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE="$PWD/../logs/active-responses.log"

#----- Analyze command -----#
if [ ${COMMAND} = "add" ]
then
  # Send control message to execd
  printf '{"version":1,"origin":{"name":"remove-threat","module":"active-response"},"command":"check_keys", "parameters":{"keys":[]}}\n'

  read RESPONSE
  COMMAND2=$(echo $RESPONSE | jq -r .command)
  if [ ${COMMAND2} != "continue" ]
  then
    echo "`date +%Y/%m/%d %H:%M:%S`" $0: $INPUT_JSON Remove threat active response aborted" >> ${LOG_FILE}
    exit 0;
  fi
fi

# Removing file
rm -f $FILENAME
if [ $? -eq 0 ]; then
  echo "`date +%Y/%m/%d %H:%M:%S`" $0: $INPUT_JSON Successfully removed threat" >> ${LOG_FILE}
else
  echo "`date +%Y/%m/%d %H:%M:%S`" $0: $INPUT_JSON Error removing threat" >> ${LOG_FILE}
fi
exit 0;
```

Figure 34 : Script pour suppression des fichiers malveillants

4. Modifiez la propriété du fichier `/var/ossec/active-response/bin/remove-threat.sh` et les permissions :

```
$ sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh
$ sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh
```

Figure 35 : Modification des permissions

5. Redémarrez l'agent Wazuh pour appliquer les changements :

```
$ sudo systemctl restart wazuh-agent
```

Figure 36 : Redémarrage Coté Serveur

1. Ajoutez les règles suivantes au fichier `/var/ossec/etc/rules/local_rules.xml` sur le serveur Wazuh. Ces règles alertent des modifications dans le repertoire `/root` détectées par les analyses FIM.

```
<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">
  <!-- Rules for Linux systems -->
  <rule id="100200" level="7">
    <if_sid>550</if_sid>
    <field name="file">/root</field>
    <description>File modified in /root directory.</description>
  </rule>
  <rule id="100201" level="7">
    <if_sid>554</if_sid>
    <field name="file">/root</field>
    <description>File added to /root directory.</description>
  </rule>
</group>
```

Figure 37: Règle d'alerte

2. Ajoutez les règles suivantes au fichier `/var/ossec/etc/rules/local_rules.xml` du serveur Wazuh pour recevoir des alertes concernant les résultats des réponses actives:

```
<group name="virustotal,">
  <rule id="100092" level="12">
    <if_sid>657</if_sid>
    <match>Successfully removed threat</match>
    <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>
  <rule id="100093" level="12">
    <if_sid>657</if_sid>
    <match>Error removing threat</match>
    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>
</group>
```

Figure 38: Règles de reception d'alerte

3. Ajoutez la configuration suivante dans le fichier `/var/ossec/etc/ossec.conf` du serveur Wazuh pour activer l'intégration avec VirusTotal. Remplacez `<VOTRE_CLÉ_API_VIRUSTOTAL>` par votre clé d'API VirusTotal. Cela permet de déclencher une requête VirusTotal chaque fois que les règles 100200 et 100201 sont activées.

```
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>4e6254a28a7a07b3d9eb43252ec07fb856c34df0a717664dd8c866a859afd7dd</api_key> <!-- Replace with your VirusTotal API key -->
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
```

Figure 39 : Integration de VirusTotal

4. Ajoutez les blocs suivants au fichier `/var/ossec/etc/ossec.conf` du serveur Wazuh. Cela active la réponse active et déclenche le script `remove-threat.sh` lorsque VirusTotal identifie un fichier comme malveillant :

```
<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.sh</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>
```

Figure 40 : Activation de réponse

b. Après La configuration du VirusTotal :

1. Le téléchargement d'un virus "eicar.com" :

```
root@osboxes:~# cd /root
root@osboxes:~# ls
eicar.com
root@osboxes:~# sudo curl -LO https://secure.eicar.org/eicar.com && ls -lah eicar.com
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100    68  100    68    0     0    169      0  --:--:-- --:--:-- --:--:--   169
-rw-r--r-- 1 root root 68 Jun  8 11:30 eicar.com
```

Figure 41 : Téléchargement du virus

2. La détection et la suppression automatique du Virus :

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jun 8, 2023 @ 16:30:20.240	004	osboxes			active-response/bin/remove-threat.sh removed threat located at /root/elcar.com	12	100092
> Jun 8, 2023 @ 16:30:19.214	004	osboxes	T1203	Execution	VirusTotal: Alert - /root/elcar.com - 63 engines detected this file	12	87105

Figure 42 : La détection et la suppression du virus

3. Et voilà le virus a été supprimer automatiquement :

```
root@osboxes:~# ls
root@osboxes:~#
```

Figure 43 : La suppression automatique "Point de vue bash"

3.4. Intégrité des fichiers

La surveillance de l'intégrité des fichiers (File Integrity Monitoring en anglais) est une technique de sécurité utilisée pour détecter les modifications non autorisées ou inattendues apportées aux fichiers d'un système informatique. Cela implique la surveillance régulière des fichiers sensibles ou critiques, tels que les fichiers système, les fichiers de configuration et les fichiers exécutables, pour détecter tout changement, qu'il soit intentionnel ou malveillant.

La surveillance de l'intégrité des fichiers repose généralement sur la comparaison des empreintes cryptographiques (hash) des fichiers avec des valeurs préalablement enregistrées. Si une différence est détectée entre l'empreinte actuelle et celle enregistrée, une alerte est déclenchée, indiquant une éventuelle violation de l'intégrité des fichiers.

a. Description de la configuration du FIM :

Nous allons activer la surveillance d'intégrité des fichiers pour le dossier "important_files", situé dans le répertoire root, afin de détecter toute modification, suppression ou ajout de fichiers.

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
  <directories realtime="yes">/root</directories>
  <directories check_all="yes" realtime="yes">/root/important_files</directories>
```

Figure 44 : Activation de la surveillance d'intégrité des fichiers

La modification du fichier file.txt : "La modification du contenu du fichier file.txt"

```
root@osboxes:~/important_files# echo "This file has been modified :)!!!" > file.txt
```

Figure 45 : Modification du fichier file.txt

b. Résultat après modification du fichier :

Jun 8, 2023 @ 17:20:57.221	004	osboxes	File modified in /root directory.	7	100200
Table	JSON	Rule			
@timestamp		2023-06-08T16:20:57.221Z			
id		rErPm4gBgqr3kHZesiOU			
agent.id		004			
agent.ip		192.168.11.149			
agent.name		osboxes			
decoder.name		syscheck_integrity_changed			
full_log		File '/root/important_files/file.txt' modified Mode: realtime Changed attributes: size,mtime,md5,sha1,sha256 Size changed from '62' to '45' Old modification time was: '1686241180', now it is '1686241257' Old md5sum was: '7a40141669fa4ea0bef3c9c00ae806e9' New md5sum is: 'a8594563868bbfd7628f0729515ad35c' Old sha1sum was: '2d23a2fb25dd05a9bd74eb840d8dea3aab6753a' New sha1sum is: 'c5aee65b288275bcd9940932f6486140e178e3' Old sha256sum was: '83808d3c510389988c77f752308ca22066d034809e036dd1d25410207e933973' New sha256sum is: '41ded0e2062f33b252c2c3dc64f97d6382c5bdf4463f088a21d0cac6245fbb26'			
id		1686241257.19728			
input.type		log			

Figure 46 : Résultat d'alert après la modification

3.5. Détection des attaques par injection SQL

a. Configuration de la surveillance des fichiers de logs Apache dans Wazuh:

Cette configuration spécifie la surveillance d'un fichier de journal d'accès Apache spécifique situé à l'emplacement `"/var/log/apache2/access.log"` et utilise le format de journal `"apache"`.

```
<ossec_config>
  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/access.log</location>
  </localfile>
</ossec_config>
```

Figure 47 : Configuration de la surveillance des fichiers log "web apache"

b. Injection SQL pour récupérer les données des utilisateurs:

La requête est envoyée à l'adresse `"http://192.168.11.149/users/"` avec un paramètre `"id"` défini comme `"SELECT++FROM+users"`. Cela indique au serveur web de récupérer les informations des utilisateurs en exécutant une requête SQL pour sélectionner toutes les colonnes () de la table `"users"`.

```
root@osboxes:~/important_files# curl -XGET "http://<UBUNTU_IP>/users/?id=SELECT+*+FROM+users";
```

Figure 48 : Injection SQL

c. **Erreur 404: Page non trouvée:**

Cela indique que la requête n'a pas été trouvée sur le serveur, ce qui a entraîné une erreur 404

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.56 (Debian) Server at 192.168.11.149 Port 80</address>
</body></html>
```

Figure 49 : Erreur 404

Conclusion Générale

Ce rapport nous a offert une précieuse opportunité d'explorer le rôle essentiel d'un système SIEM/XDR au sein d'un SOC, ainsi que les outils sophistiqués utilisés par les équipes Blue Team pour surveiller, gérer et répondre aux événements de sécurité. Nous avons comparé des solutions open source et commerciales pour comprendre leurs avantages et leurs limitations. Nous avons constaté que le domaine de la sécurité informatique regorge de connaissances à approfondir et de défis passionnants à relever.

Dans un environnement SIEM/XDR, des agents ou des capteurs collectent des données sur les terminaux et les réseaux, qui sont ensuite analysées pour détecter les activités suspectes et générer des alertes. Nous avons été impressionnés par la capacité de ces solutions à identifier les schémas d'activité suspects et les indicateurs de compromission potentiels.

Cependant, nous sommes conscients qu'il reste encore beaucoup à apprendre pour maîtriser pleinement ces technologies et les intégrer de manière efficace dans un environnement de sécurité. Ce rapport a renforcé notre conviction que la sécurité informatique est un domaine en constante évolution, nécessitant une expertise approfondie, une veille technologique constante et une volonté d'apprentissage permanent.

Nous sommes impatients de poursuivre notre parcours dans ce domaine passionnant et de relever les défis futurs avec détermination et persévérance.

Bibliographie

<https://wazuh.com/>

https://www.youtube.com/watch?v=Hq58_yGJwHk&t

<https://www.youtube.com/watch?v=2HMo4h7elAA>

<https://openai.com/blog/chatgpt> <https://owasp.org/www-project-top-ten/>

<https://www.virustotal.com/gui/> <https://nmap.org>

Fin.