LAB DE DÉTECTION D'INTRUSION & ANALYSE DE MENACES AVEC WAZUH

Mise en œuvre de Wazuh : Détection, Analyse de logs et Réponse aux incidents (Windows/Linux)

Aurélien LIN

Décembre 2025

TABLE DES MATIÈRES

01 INTRODUCTION

1.1. Objectifs du laboratoire de surveillance

Ce projet s'inscrit dans une démarche d'apprentissage pratique des métiers de la cybersécurité, plus précisément de l'analyse SOC (Security Operations Center). L'objectif principal est de construire un environnement de détection d'intrusion capable de surveiller des systèmes hétérogènes (Windows et Linux).

Les buts spécifiques de ce laboratoire sont :

- La centralisation des événements : Regrouper les logs de sécurité provenant de différentes sources sur une plateforme unique.
- La détection de menaces réelles : Simuler des attaques (Brute Force, Malware, Scan de ports) pour vérifier la réactivité du système.
- La réponse aux incidents : Établir des procédures claires (Playbooks) pour neutraliser les menaces détectées.

1.2. Présentation de la solution SIEM Wazuh

Pour ce laboratoire, le choix s'est porté sur Wazuh, une solution de sécurité open-source de référence intégrant des capacités de SIEM (Security Information and Event Management) et de XDR (Extended Detection and Response).

Wazuh a été sélectionné pour ses fonctionnalités clés :

- Analyse des logs : Collecte et analyse automatique des journaux d'événements système.
- Surveillance d'intégrité (FIM) : Détection en temps réel des modifications sur des fichiers critiques.
- Détection d'anomalies : Identification de comportements suspects grâce à un moteur de règles puissant.
- Évaluation de la conformité : Vérification de la configuration de sécurité des agents déployés.

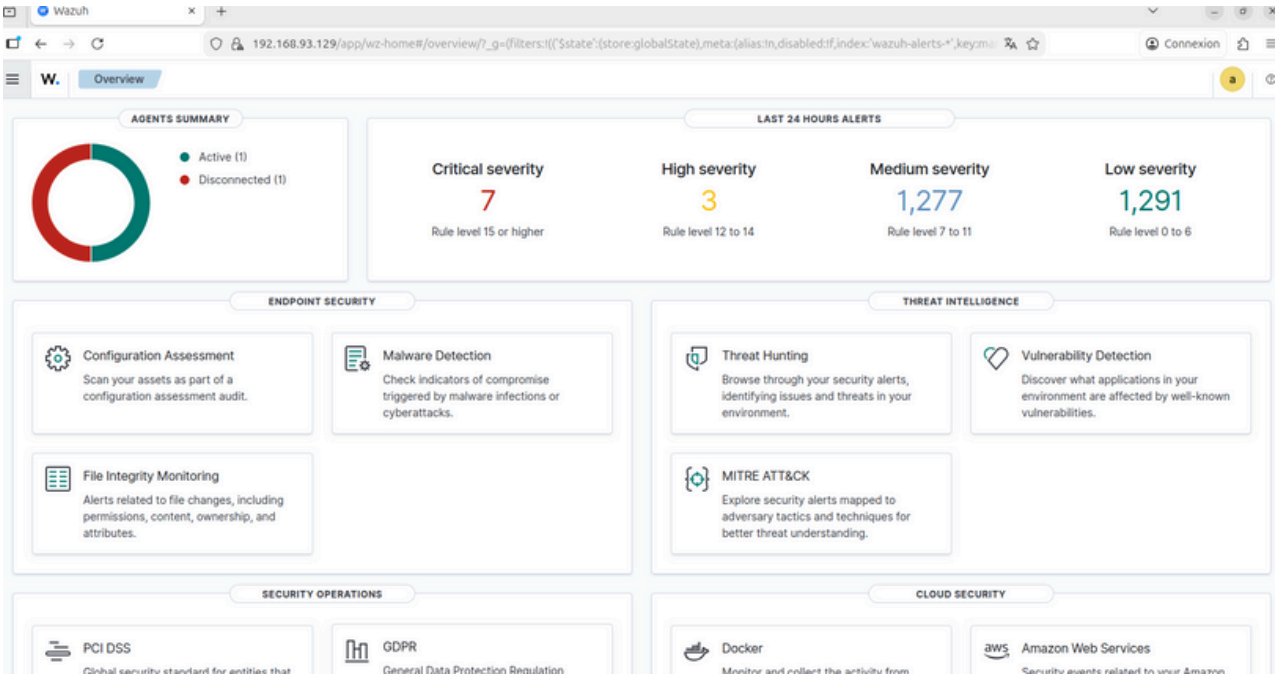


Figure 1 : Interface de supervision centralisée (Dashboard Wazuh).

02 ARCHITECTURE TECHNIQUE

2.1. Environnement de virtualisation VMware

L'intégralité du laboratoire est déployée sur VMware Workstation. Les machines sont configurées sur un réseau en mode NAT, ce qui permet d'isoler le laboratoire du réseau physique tout en permettant aux machines de communiquer entre elles via un sous-réseau privé.

- SRV-WAZUH (Ubuntu) : 4 Go RAM | 2 vCPU | 50 Go SSD.
- WIN11-TARGET (Windows) : 4 Go RAM | 2 vCPU | 64 Go SSD.
- SRV-LINUX-TEST (Clone) : 4 Go RAM | 2 vCPU | 50 Go SSD.

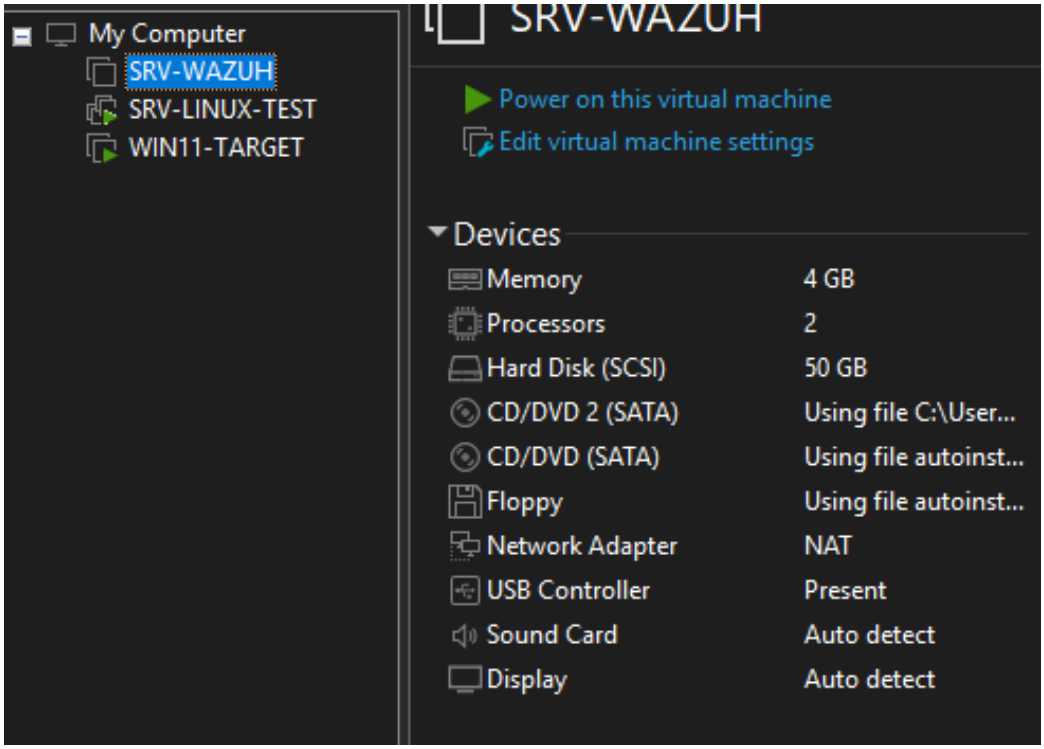


Figure 2 : Détails de la configuration matérielle sous VMware.

2.2. Inventaire des actifs

Nom de l'actif	Système d'Exploitation	Rôle / Fonction
SRV-WAZUH	Ubuntu 24.04 LTS	Manager : Serveur central d'analyse et de contrôle.
WIN11-TARGET	Windows 11	Endpoint Windows : Poste client surveillé.
SRV-LINUX-TEST	Ubuntu 24.04 LTS	Endpoint & Outil d'attaque : Machine utilisée pour simuler les menaces et tester la détection Linux.

2.3. Topologie réseau et flux de communication

La communication est centralisée vers le manager, comme le confirme l'interface de supervision :

- Flux d'agent (Ports 1514/1515) : Transmission sécurisée des logs depuis les endpoints vers l'IP 192.168.93.129.
- Interface Dashboard (Port 443) : Accès à l'interface de gestion via navigateur.
- Sécurité : Le tableau de bord affiche un résumé des agents actifs et permet de visualiser la sévérité des alertes détectées.

03 DÉPLOIEMENT DU SIEM

3.1. Installation du Wazuh Manager sur Ubuntu

Le serveur central a été déployé sur la machine SRV-WAZUH via le script d'installation automatique. Cette méthode permet d'installer de manière cohérente l'indexeur, le serveur et le dashboard sur un nœud unique.

Commande utilisée : curl -sO https://packages.wazuh.com/4.x/wazuh-install.sh && sudo bash wazuh-install.sh -a.

Validation : Après l'exécution, l'accès à l'interface a été vérifié via l'adresse https://192.168.93.129.

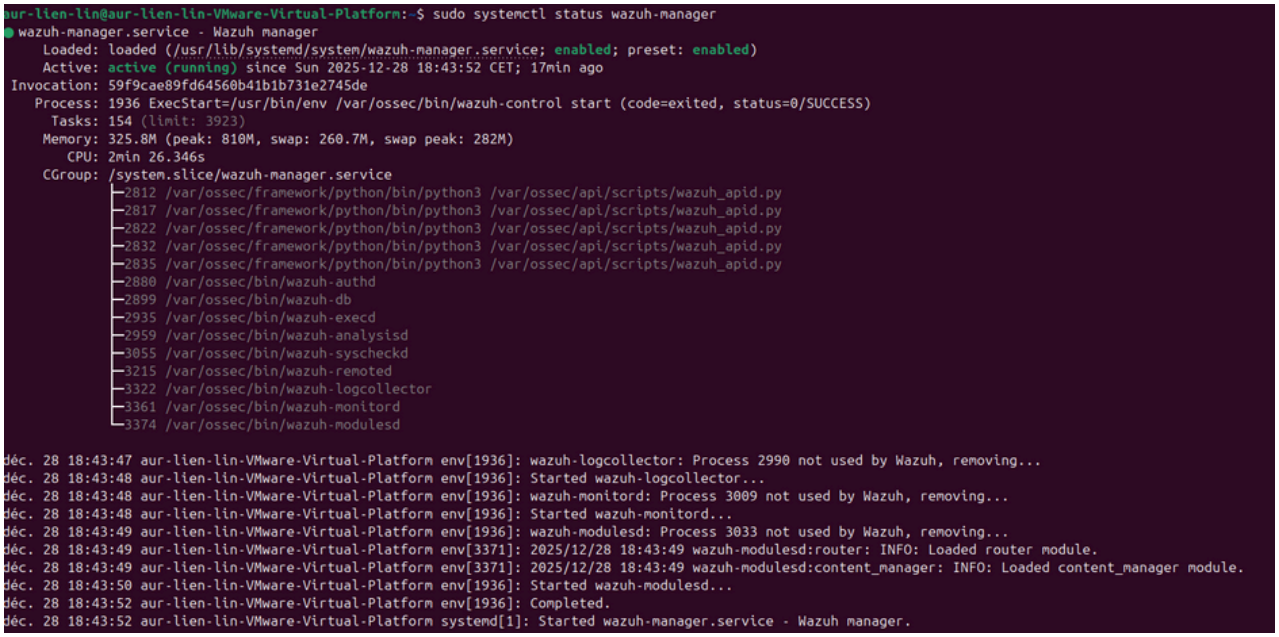


Figure 3 : Confirmation du statut opérationnel "active (running)" du Manager Wazuh.

3.2. Configuration et déploiement des agents

Pour intégrer les machines cibles dans le périmètre de surveillance, le déploiement s'est fait depuis le menu "Deploy new agent" du dashboard.

- Agent Windows (WIN11-TARGET) : Téléchargement du package MSI et installation via PowerShell avec les paramètres d'IP du manager.

- Agent Linux (SRV-LINUX-TEST) : Installation via le dépôt APT et configuration du service wazuh-agent.

La réussite du déploiement est confirmée par l'apparition des deux agents avec le statut "Active" dans la console.

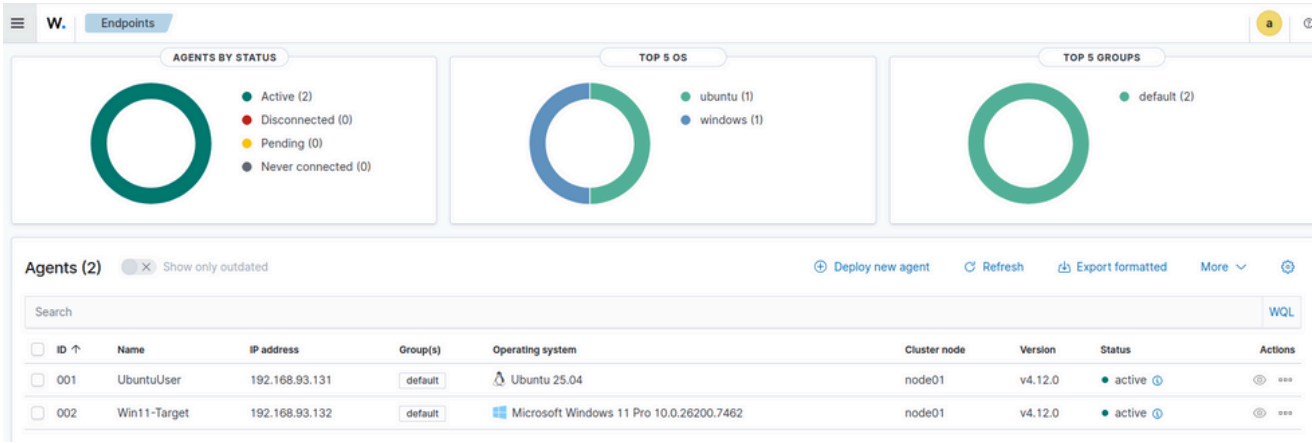


Figure 4 : Inventaire des agents actifs et leurs systèmes d'exploitation respectifs.

04 SCÉNARIOS D'ATTAQUES ET DÉTECTION

Dans cette phase, nous simulons des comportements malveillants pour vérifier l'efficacité des règles de détection de Wazuh.

4.1. Environnement Linux (Ubuntu)

4.1.1. Attaque n°1 : Brute Force SSH avec Hydra

Objectif : Tester la capacité du SIEM à détecter une tentative d'intrusion par force brute sur le service SSH.

Action : Utilisation de l'outil Hydra depuis la machine attaquante pour tenter de deviner le mot de passe du compte "root".

Exécution : La commande génère un flux rapide de tentatives de connexion (ici 1000 tentatives avec 4 tâches en parallèle) vers la cible 192.168.93.129.

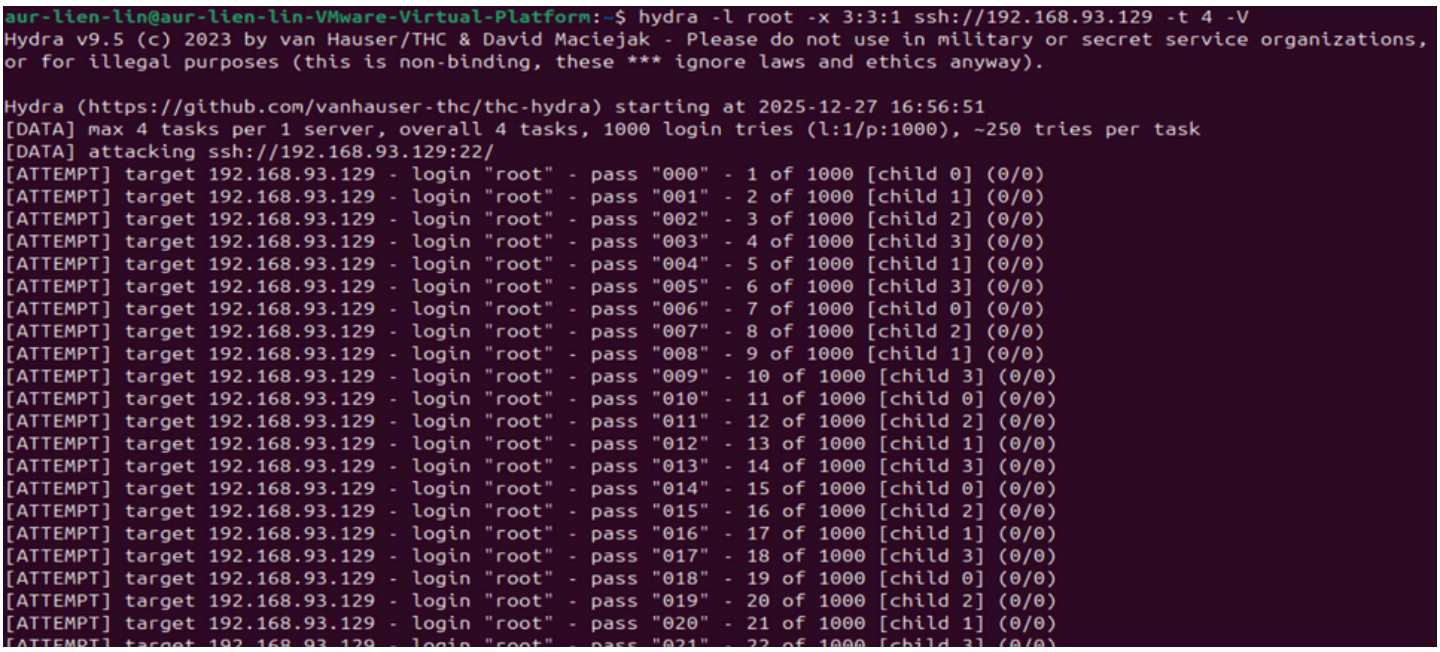


Figure 5 : Lancement de l'attaque Hydra simulant 1000 tentatives de connexion SSH.

Détection Wazuh : Le manager identifie immédiatement une explosion du nombre d'échecs d'authentification.

Analyse du Dashboard : La console affiche une accumulation massive d'alertes de niveau 5 ("sshd: authentication failed"). On observe un pic d'activité critique dans les graphiques de "Threat Hunting", avec 88 échecs d'authentification comptabilisés en un temps très court.

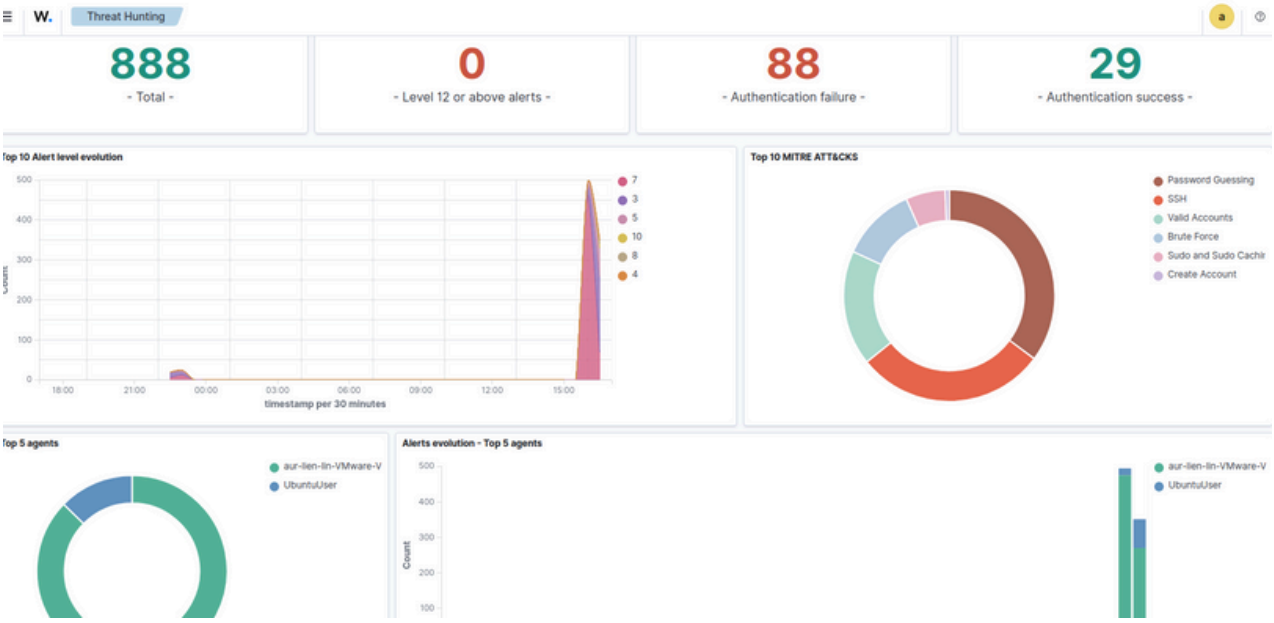
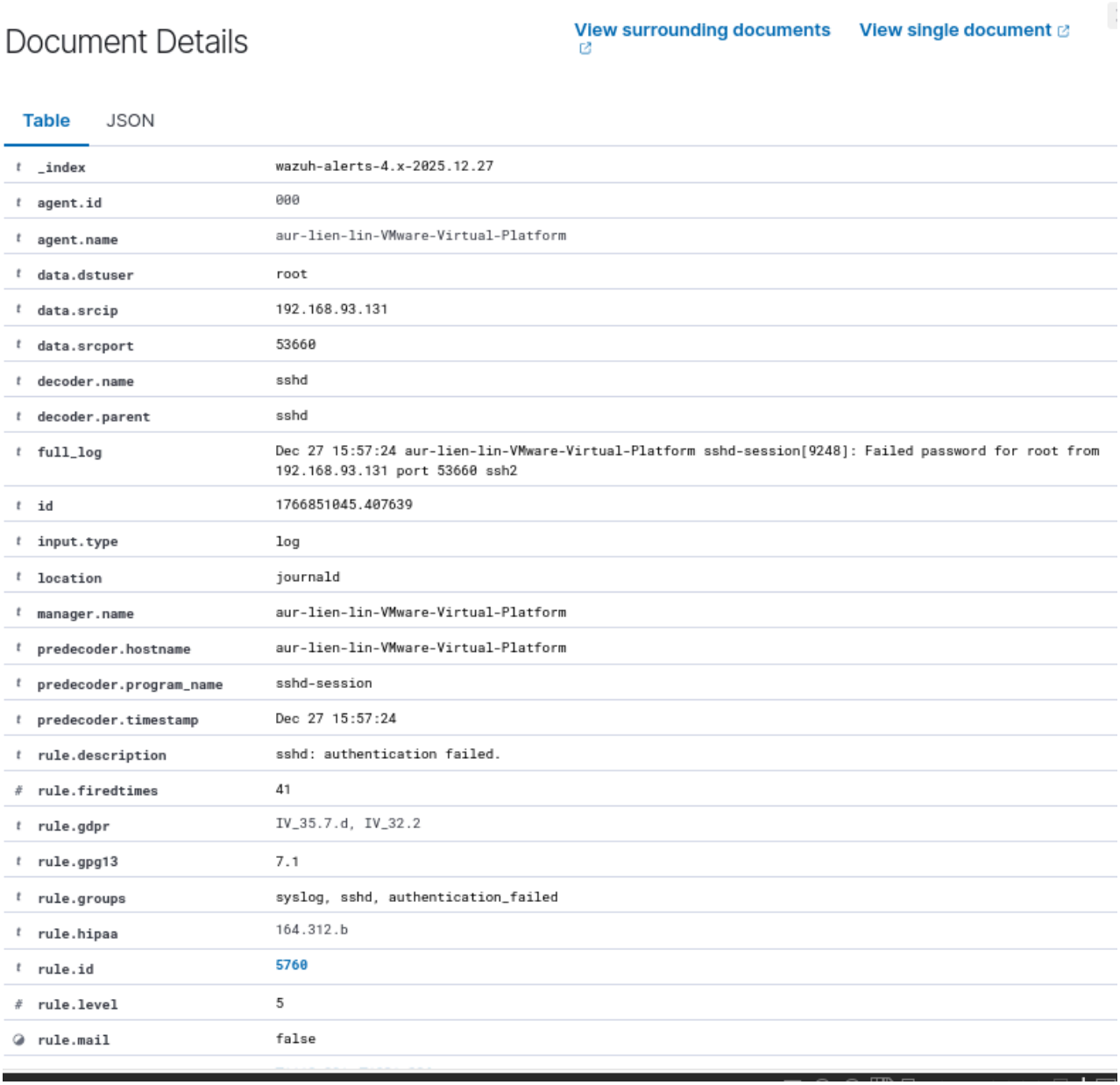


Figure 6 : Dashboard Wazuh montrant le pic d'échecs d'authentification (88 hits)

Note d'analyse : Bien que l'outil Hydra ait généré 1000 tentatives, le Dashboard affiche 88 alertes distinctes. Cet écart s'explique par le mécanisme d'agrégation de Wazuh qui regroupe les événements répétitifs pour optimiser la visibilité et éviter le 'flood' de la console de supervision.

Détail technique de l'alerte : En inspectant l'événement précis (ID de règle 5760), Wazuh extrait des informations cruciales pour l'investigation : l'IP source de l'attaquant (192.168.93.131), l'utilisateur ciblé (root) et le port utilisé (53660).



Document Details

View surrounding documents | View single document

Table | JSON

t _index	wazuh-alerts-4.x-2025.12.27
t agent.id	000
t agent.name	aur-lien-lin-VMware-Virtual-Platform
t data.dstuser	root
t data.srcip	192.168.93.131
t data.srcport	53660
t decoder.name	sshd
t decoder.parent	sshd
t full_log	Dec 27 15:57:24 aur-lien-lin-VMware-Virtual-Platform sshd-session[9248]: Failed password for root from 192.168.93.131 port 53660 ssh2
t id	1766851045.407639
t input.type	log
t location	journald
t manager.name	aur-lien-lin-VMware-Virtual-Platform
t predecoder.hostname	aur-lien-lin-VMware-Virtual-Platform
t predecoder.program_name	sshd-session
t predecoder.timestamp	Dec 27 15:57:24
t rule.description	sshd: authentication failed.
# rule.firedtimes	41
t rule.gdpr	IV_35.7.d, IV_32.2
t rule.gpg13	7.1
t rule.groups	syslog, sshd, authentication_failed
t rule.hipaa	164.312.b
t rule.id	5760
# rule.level	5
rule.mail	false

Figure 7 : Analyse granulaire de l'alerte identifiant l'adresse IP de l'attaquant.

4.1.2. Attaque n°2 : Création d'un utilisateur suspect

Objectif : Simuler une tentative de persistance sur le système en créant un nouveau compte utilisateur sans autorisation.

Action : Exécution de la commande de création d'utilisateur sur l'agent Ubuntu.

Exécution : sudo adduser hacker_soc

Analyse de la détection : Le module d'analyse de Wazuh identifie en temps réel une série d'événements critiques liés à la gestion des comptes. Le Dashboard "Events" regroupe quatre alertes distinctes survenant simultanément :

- New group added to the system (ID 5901 - Niveau 8).
- New user added to the system (ID 5902 - Niveau 8).
- Information from the user was changed (ID 5904 - Niveau 8).

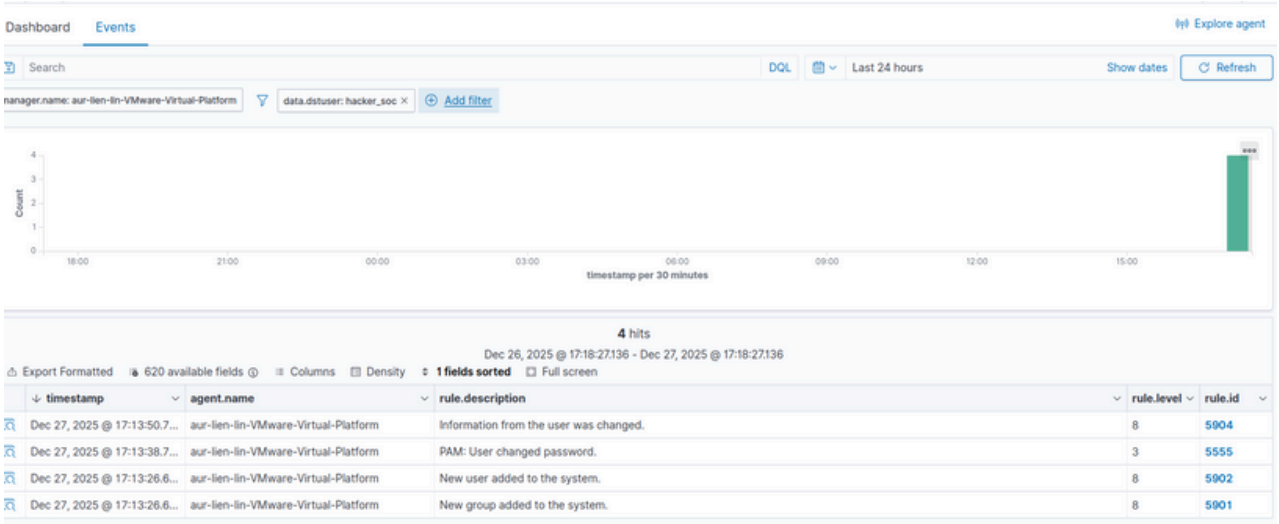


Figure 8 : Chronologie des événements Wazuh lors de la création de l'utilisateur "hacker_soc"

Preuve granulaire : L'inspection détaillée de l'événement permet de confirmer l'identité de l'intrus. Le SIEM extrait précisément le nom de l'utilisateur créé (hacker_soc), son UID/GID (1001), ainsi que le binaire système utilisé pour l'action (useradd).

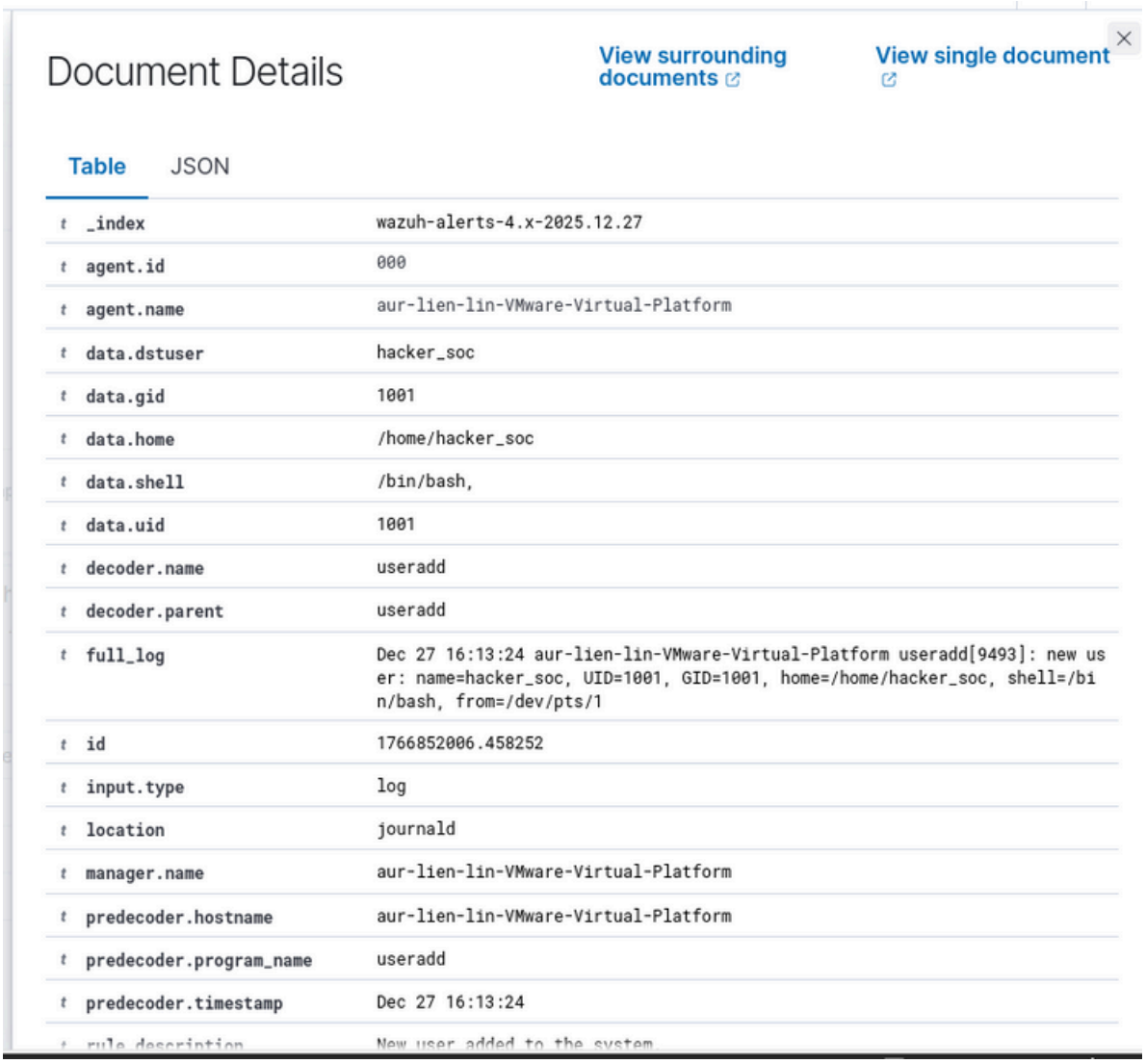


Figure 9 : Document Details confirmant la création du compte suspect et ses attributs système.

4.1.3. Attaque n°3 : Suppression des logs avec la commande truncate.

Objectif : Simuler une tentative de dissimulation d'activité (Defense Evasion) en vidant les journaux d'authentification pour effacer les traces de l'intrusion.

Action : Exécution d'une commande de remise à zéro du fichier de log sur l'agent Ubuntu avec les privilèges ROOT.

Exécution : sudo truncate -s 0 /var/log/auth.log

Analyse de la détection : Le SIEM identifie une anomalie critique liée à l'intégrité des journaux système. Le Dashboard "Events" permet de corréler cette action avec les étapes précédentes de l'attaque :

- Détection de la commande : Wazuh enregistre l'exécution de la commande truncate visant le fichier de logs.

- Chronologie suspecte : L'alerte survient immédiatement après les succès de la commande sudo (ID 5402), ce qui confirme une tentative de "nettoyage" après l'obtention des droits ROOT.

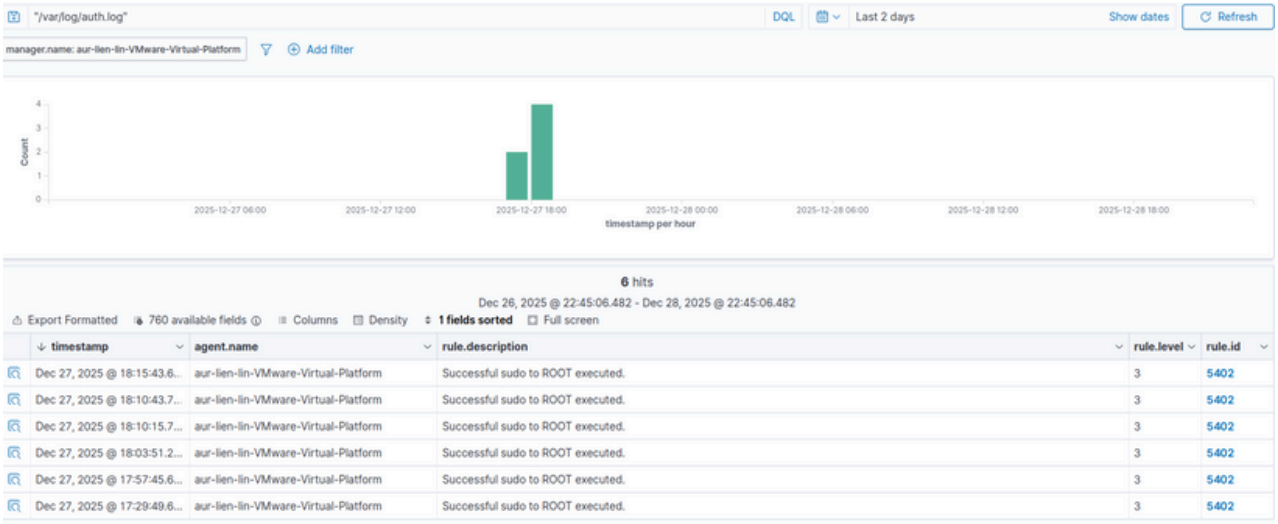


Figure 10 : Timeline montrant l'enchaînement logique entre l'accès ROOT et l'effacement des logs.

Preuve granulaire : L'inspection détaillée de l'événement permet de confirmer la précision de la détection. Le SIEM extrait le chemin complet du binaire utilisé (/usr/bin/truncate), l'argument utilisé (-s 0) pour vider le fichier, ainsi que l'utilisateur responsable de l'action (aur-lien-lin).

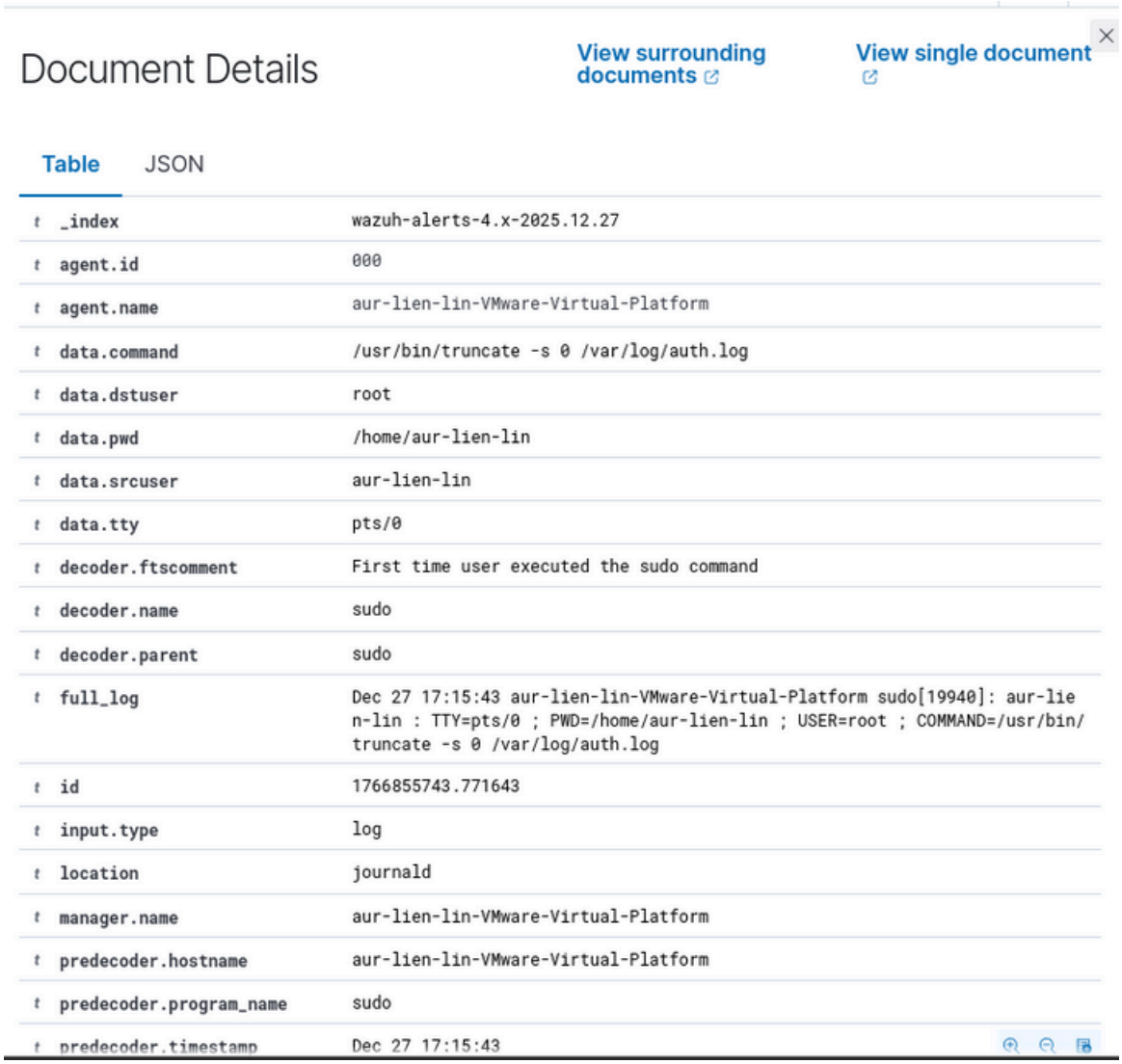


Figure 11 : Détails techniques extraits par Wazuh montrant la commande exacte de suppression des traces.

4.2. Environnement Windows (Windows 11)

4.2.1. Attaque n°4 : Scan de ports agressif (via Nmap)

Objectif : Simuler une phase de reconnaissance agressive pour identifier l'état de la cible Windows 11 et ses services exposés.

Action : Exécution d'un scan de ports depuis la machine attaquante vers l'agent Windows.

Exécution : Utilisation de la commande nmap -Pn 192.168.93.132 pour contourner le blocage du ping (ICMP) et forcer l'analyse de la cible.

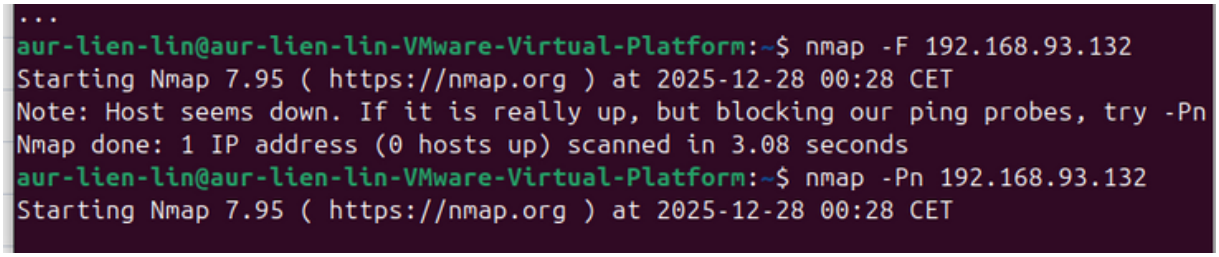


Figure 12 : Lancement du scan réseau Nmap vers la cible Windows 11.

Analyse de la détection : Le SIEM Wazuh identifie immédiatement un flux de trafic anormal sur l'agent Win11-Target. Le dashboard "Discover" affiche un volume massif de 714 alertes générées en un temps très court, ce qui est caractéristique d'un scan automatisé.

Preuve granulaire : L'inspection des logs système sur Wazuh permet de confirmer que les événements réseau sont générés sur l'agent Windows ciblé (192.168.93.132) à la suite d'un scan de ports initié depuis la machine attaquante. On observe notamment des requêtes liées à des services Windows (comme le protocole WPAD), ce qui confirme une phase de reconnaissance active visant les ports et services exposés du système.

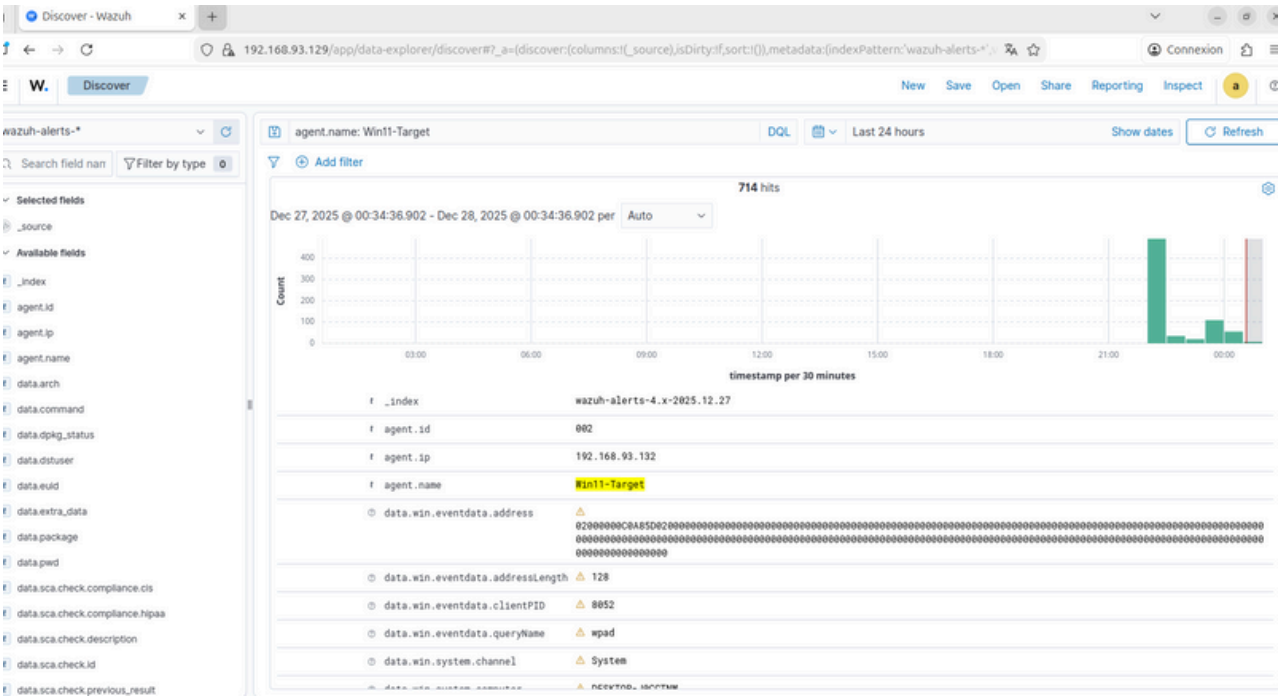


Figure 13 : Vue globale du SIEM montrant le pic de 714 événements suite au scan Nmap.

4.2.2. Attaque n°5 : Brute Force RDP

Objectif : Simuler une tentative d'accès par force brute sur un compte utilisateur local pour tester la réactivité du SIEM face à des échecs d'authentification massifs.

Action : Utilisation d'un script PowerShell pour automatiser des tentatives de connexion avec des identifiants erronés.

Exécution : Exécution d'une boucle for ciblant l'utilisateur "Pirate" sur l'interface de bouclage locale, générant 20 tentatives de connexion infructueuses en quelques secondes.

```
PS C:\Users\linri> for ($i=1; $i -le 20; $i++) { net use \\127.0.0.1\c$ /user:Pirate "FauxPass$i" 2>$null }
```

Figure 14 : Script PowerShell simulant l'attaque par Brute Force sur l'utilisateur "Pirate".

Analyse de la détection : Wazuh remonte immédiatement des alertes de niveau 10. Le SIEM décode les journaux d'audit de Windows et identifie la répétition anormale de l'événement 4625 (An account failed to log on) sur l'agent Win11-Target.

Preuve granulaire : L'inspection des champs data.win.eventdata permet de confirmer les paramètres de l'attaque : l'adresse IP source (127.0.0.1), le nom du compte visé (Pirate) et le processus d'authentification utilisé (NtLmSsp).

Figure 15 : Dashboard Wazuh affichant les alertes critiques liées aux échecs de connexion.

4.2.3. Attaque n°6 : Malware (Test EICAR)

Objectif : Vérifier la capacité de Wazuh à centraliser les alertes de sécurité provenant de l'antivirus natif (Windows Defender) lors de l'introduction d'un fichier malveillant.

Action : Création d'un fichier texte contenant la signature de test EICAR (European Institute for Computer Antivirus Research).

Exécution : Un fichier nommé test.txt est créé sur le bureau de la VM Windows avec la chaîne de caractères spécifique reconnue par tous les moteurs antivirus.

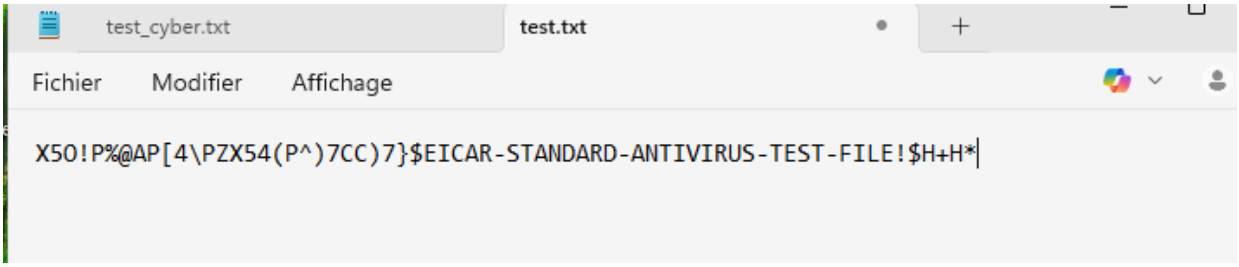


Figure 16 : Création du fichier de test EICAR sur l'agent Windows.

Analyse de la détection : L'antivirus Microsoft Defender identifie immédiatement la menace et génère une alerte système. Cette information est instantanément transmise au manager Wazuh via l'agent installé sur la machine.

- **Visibilité globale** : Le dashboard principal affiche une augmentation des alertes de "High severity" (niveau 12) liées à la détection de menaces.

- **Intégration Endpoint Security** : Le module "Malware Detection" de Wazuh confirme la surveillance active des indicateurs de compromission.

Preuve granulaire : L'examen des logs détaillés montre que le SIEM récupère non seulement l'alerte, mais aussi des métadonnées cruciales comme les empreintes numériques (Hashes MD5/SHA256) du fichier suspect. Cela permet à un analyste de vérifier l'intégrité du fichier sur des bases de données de menaces externes.

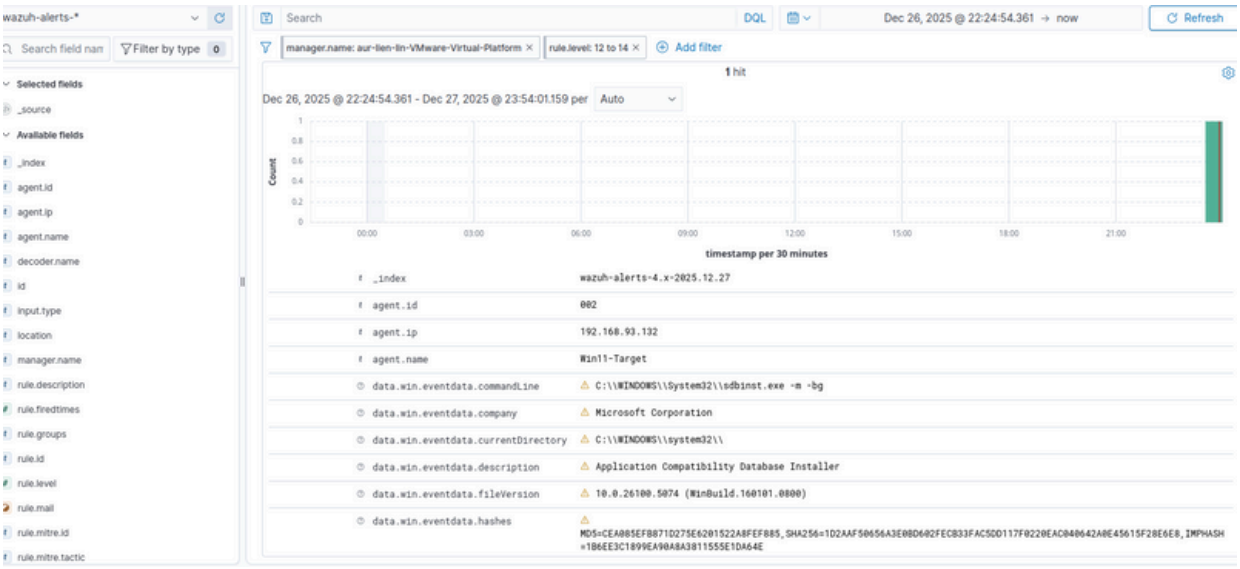


Figure 17 : Isolation des alertes de niveau 12 sur l'agent Win11-Target

05 PLAYBOOKS ET RÉPONSE AUX INCIDENTS

5.1. Analyse des alertes et corrélation des événements

L'analyse consiste à regrouper les 6 attaques simulées pour comprendre la logique de l'intrus, de la reconnaissance à l'effacement des traces.

Analyse de la chaîne d'attaque : En corrélant les alertes, on observe que les scans réseau et le brute force ont servi de porte d'entrée avant l'escalade de privilèges. Le dashboard de Wazuh permet de visualiser cette montée en puissance de l'attaque en temps réel.

Validation de l'intrusion (Sudo) : La preuve concrète du succès de l'attaquant se trouve dans les logs de l'ID de règle 5402. On y voit 6 exécutions réussies de la commande sudo vers le compte ROOT, confirmant que le serveur Linux a été totalement compromis.

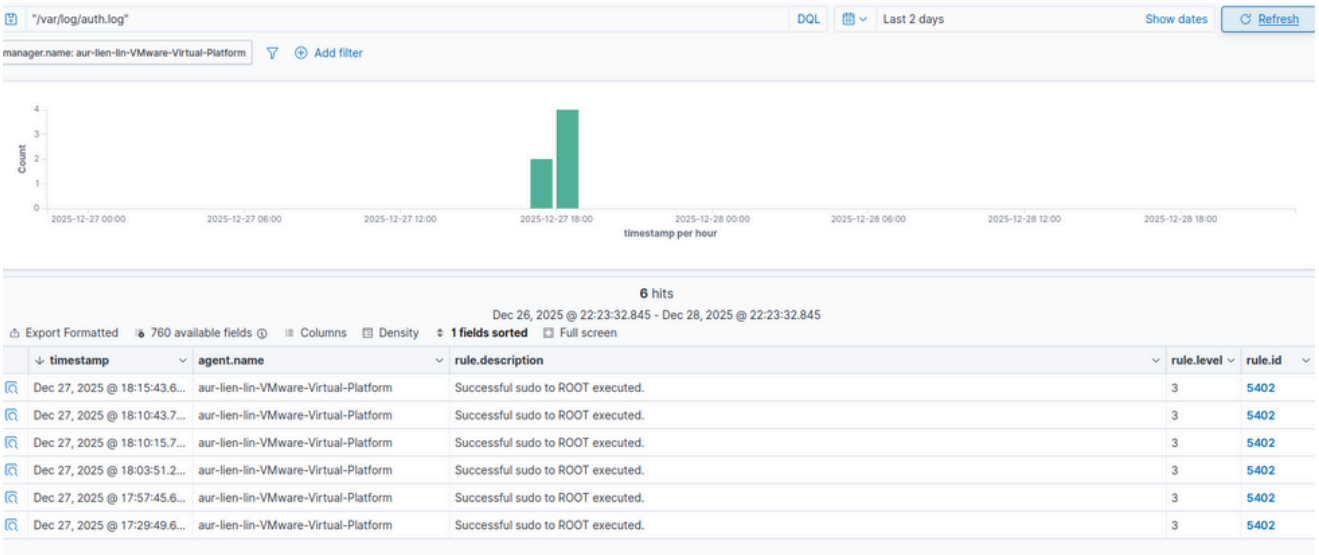


Figure 18 : Journalisation des exécutions sudo prouvant la compromission du serveur.

Mapping MITRE ATT&CK : Pour classifier les 6 vecteurs d'attaque, j'ai utilisé la vue MITRE qui identifie clairement les techniques de Credential Access et de Persistence. Cette vue d'ensemble est indispensable pour prioriser la réponse à l'incident.

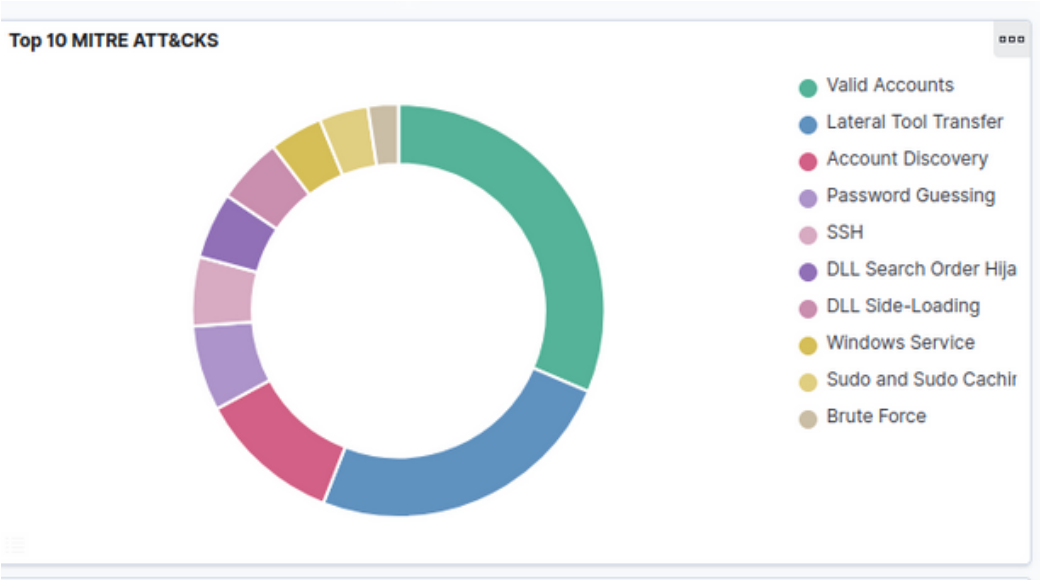


Figure 19 : Classification des menaces détectées selon le référentiel MITRE.

5.2. Procédures de remédiation et bonnes pratiques de durcissement

Suite aux différentes simulations effectuées, j'ai identifié les mesures correctives nécessaires pour sécuriser l'infrastructure et limiter l'impact d'une future attaque.

Réponse au Brute Force (Accès SSH) : La première action est de bloquer l'adresse IP de l'attaquant (192.168.93.131) identifiée dans les logs. Pour un durcissement durable, il est préconisé de désactiver l'authentification par mot de passe au profit de clés SSH sécurisées.

Contrôle des privilèges (Sudo) : La détection de succès suspects sur la règle 5402 doit déclencher un audit immédiat des droits de l'utilisateur concerné. Il est crucial d'appliquer le principe du moindre privilège pour éviter qu'un compte compromis ne devienne ROOT.

Protection contre l'évasion (Logs) : Pour contrer l'effacement des traces (commande truncate), la recommandation est d'externaliser les journaux vers un serveur de logs distant. Cela garantit que les preuves de l'attaque restent disponibles même si l'attaquant nettoie la machine locale.

Automatisation via Active Response : L'un des points clés est l'utilisation du module de réponse automatique de Wazuh pour bannir une IP hostile dès qu'un comportement anormal est détecté, réduisant ainsi le temps d'exposition de la machine.

06 CONCLUSION

Ce projet de laboratoire a permis de simuler un environnement SOC (Security Operations Center) complet et fonctionnel. La mise en place de Wazuh sur une architecture virtualisée a démontré l'efficacité de cet outil pour la détection d'intrusions en temps réel.

Les tests d'attaques réalisés ont prouvé que :

- Le SIEM est capable d'identifier des tentatives de Brute Force SSH immédiatement.

- La surveillance d'intégrité des fichiers permet de détecter des actions suspectes, comme la suppression de logs ou la modification de fichiers système.

Perspectives : Ce laboratoire pourrait être amélioré en intégrant des règles de réponse active (Active Response) pour bloquer automatiquement les adresses IP malveillantes dès la détection d'une attaque, renforçant ainsi la posture de sécurité de l'infrastructure.