École Normale Supérieure de l'Enseignement Technique (ENSET)

Virtualisation et Cloud Computing

Atelier Sécurité des Endpoints et Supervision SIEM

Étude de cas multi-OS (Linux & Windows)

Réalisé par : Machnaoui Abdellatif

Encadrant : Prof. Azeddine KHIAT

Année universitaire : 2025–2026

Table des matières

Chapitre 1

Introduction

Dans le contexte actuel marqué par la multiplication des cyberattaques et la complexité croissante des systèmes d'information, la mise en place de solutions de supervision de sécurité est devenue indispensable. Les entreprises modernes s'appuient sur des plateformes combinant les approches **SIEM** (Security Information and Event Management) et **EDR** (Endpoint Detection and Response) afin de détecter, analyser et répondre efficacement aux incidents de sécurité.

Ce projet s'inscrit dans cette logique et consiste à mettre en œuvre une plateforme complète basée sur **Wazuh**, déployée dans un environnement Cloud AWS, permettant la supervision simultanée de systèmes **Linux** et **Windows**. L'objectif principal est d'illustrer, à travers des scénarios réalistes, le fonctionnement d'un SOC moderne et la corrélation d'événements de sécurité en temps réel.

Chapitre 2

Chapitre 1 : SIEM, EDR et présentation de Wazuh

2.0.1 Introduction à la supervision de la sécurité

La sécurité des systèmes d'information modernes ne peut plus se limiter à des mécanismes préventifs tels que les pare-feu ou les antivirus traditionnels. Les attaques sont devenues plus complexes, persistantes et souvent difficiles à détecter en temps réel. Dans ce contexte, les organisations ont besoin de solutions capables de **collecter**, **corréler**, **analyser** et **interpréter** de grandes quantités d'événements de sécurité provenant de sources hétérogènes.

C'est dans cette optique qu'apparaissent les solutions de type **SIEM** et **EDR**, aujourd'hui considérées comme des briques fondamentales d'un **Security Operations Center (SOC)**.

2.0.2 SIEM : Security Information and Event Management

Un **SIEM** (Security Information and Event Management) est une plateforme centralisée dont l'objectif principal est la gestion des événements de sécurité. Elle permet de collecter des logs provenant de multiples sources telles que :

— Systèmes d'exploitation (Linux, Windows),
— Équipements réseau (firewalls, routeurs, switches),
— Applications métiers,
— Services Cloud.

Le SIEM assure plusieurs fonctions essentielles :

— **Collecte centralisée des logs**,
— **Normalisation** des événements,
— **Corrélation** de plusieurs événements entre eux,
— **Détection d'incidents** via des règles ou des signatures,

— **Visualisation** via des tableaux de bord.

Grâce à ces capacités, un SIEM permet aux analystes SOC de détecter des comportements anormaux tels que des tentatives de bruteforce, des accès non autorisés ou des escalades de privilèges.

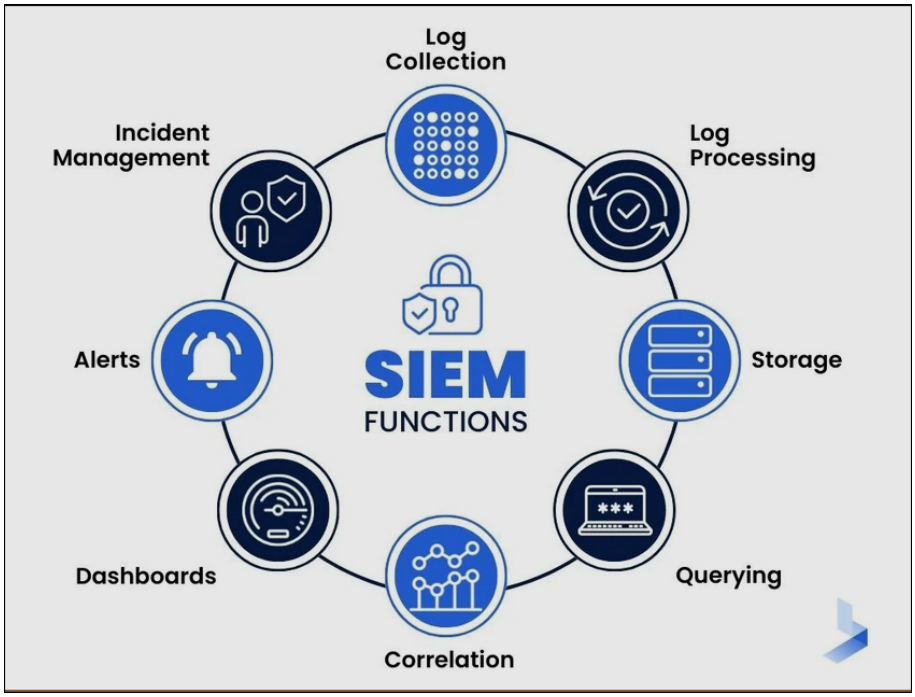

FIGURE 2.1 – Principe de fonctionnement d'un SIEM

Cependant, le SIEM présente une limite importante : il repose principalement sur l'analyse des logs et ne fournit pas toujours une visibilité fine sur le comportement interne des endpoints.

2.0.3 EDR : Endpoint Detection and Response

Un **EDR** (Endpoint Detection and Response) est une solution dédiée à la surveillance avancée des postes de travail et des serveurs. Contrairement au SIEM, l'EDR se concentre directement sur les **endpoints** et analyse en profondeur les actions réalisées sur le système.

Les fonctionnalités clés d'un EDR incluent :

— Surveillance des processus,

— Analyse des comportements utilisateurs,

— Détection d'activités malveillantes,

— Réponse aux incidents (isolation, blocage, alerte).

L'EDR permet par exemple de détecter des tentatives de persistance, des mouvements latéraux ou l'exécution de commandes suspectes, même lorsque celles-ci ne génèrent pas immédiatement des logs classiques.

2.0.4 Wazuh : une plateforme SIEM et EDR open-source

Wazuh est une plateforme open-source qui combine les capacités d'un SIEM et d'un EDR au sein d'une architecture unifiée. Elle repose sur un modèle **agent / serveur** et permet la supervision de systèmes Linux, Windows et macOS.

Les composants principaux de Wazuh sont :

— **Wazuh Agent** : installé sur les endpoints,
— **Wazuh Manager** : collecte et analyse les événements,
— **Indexer** : stockage et recherche des données,
— **Dashboard** : interface de visualisation.

Wazuh offre notamment :

— File Integrity Monitoring (FIM),
— Détection d'intrusions,
— Surveillance des comptes utilisateurs,
— Corrélation basée sur MITRE ATT&CK,
— Capacités EDR enrichies (Sysmon, audit Windows).

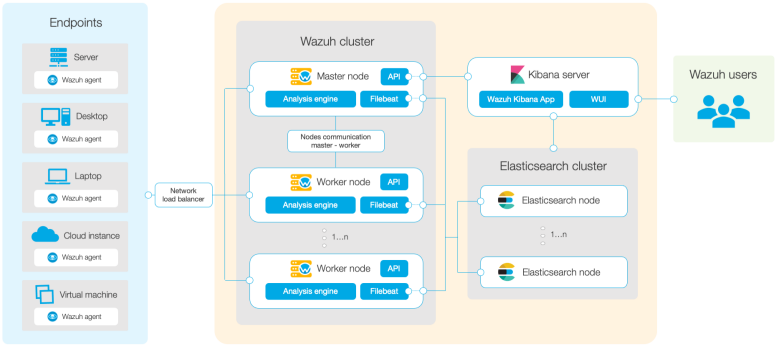

FIGURE 2.2 – Architecture logique de la plateforme Wazuh

Ainsi, Wazuh constitue une solution complète et accessible pour la mise en place d'un SOC moderne, ce qui justifie son choix dans le cadre de ce projet.

Chapitre 3

Chapitre 2 : Mise en œuvre pratique de la plateforme Wazuh

3.0.1 Objectifs du laboratoire

L'objectif de ce laboratoire est de déployer une plateforme SIEM/EDR fonctionnelle dans un environnement Cloud réel et de démontrer sa capacité à détecter et analyser des événements de sécurité sur des systèmes Linux et Windows.

3.0.2 Architecture du laboratoire

Le laboratoire a été déployé sur AWS Learner Lab et repose sur une architecture simple mais réaliste, similaire à celle utilisée dans de nombreuses entreprises.

— Une instance EC2 Ubuntu hébergeant Wazuh All-in-One,
— Une instance EC2 Ubuntu représentant un poste Linux,
— Une instance EC2 Windows Server 2025 représentant un environnement entreprise.

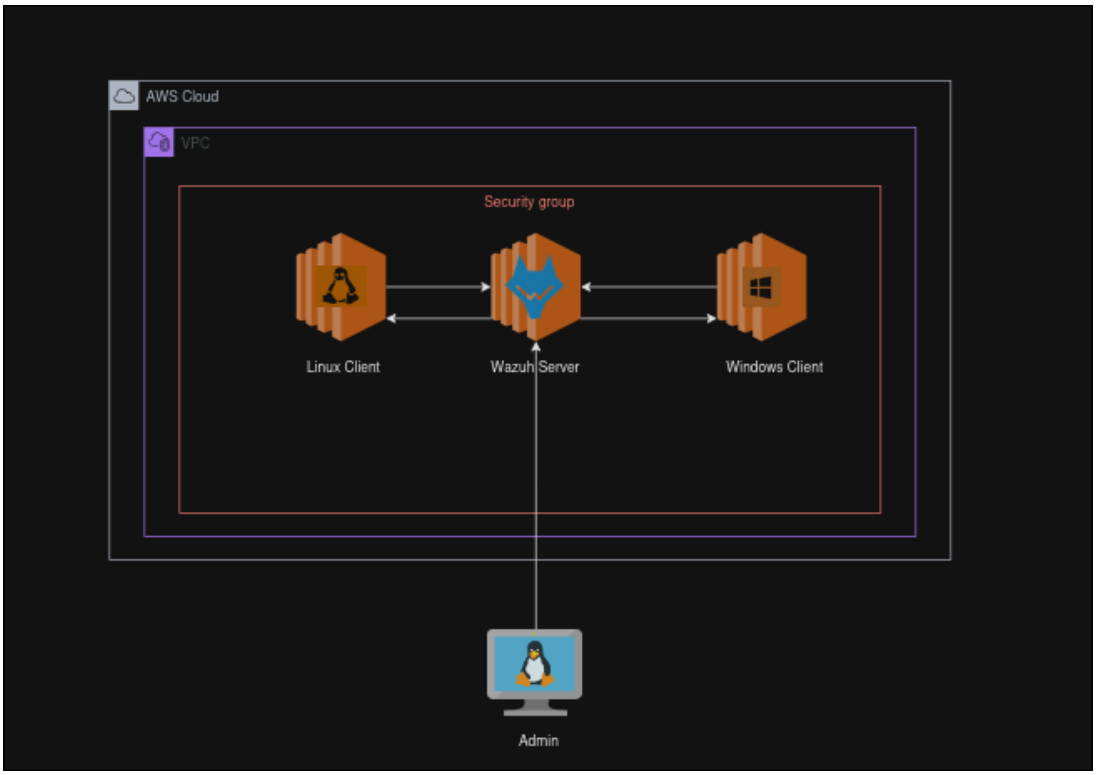

FIGURE 3.1 – Architecture du laboratoire Wazuh sur AWS

Toutes les instances sont déployées dans le même VPC afin de garantir la communication sécurisée entre les agents et le serveur Wazuh.

3.0.3 Déploiement du serveur Wazuh

Le serveur Wazuh a été installé sur une instance Ubuntu 22.04 à l'aide du script officiel All-in-One, permettant de déployer l'ensemble des composants nécessaires.

```
1 sudo apt update && sudo apt upgrade -y
2 curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
3 sudo bash wazuh-install.sh -a
```

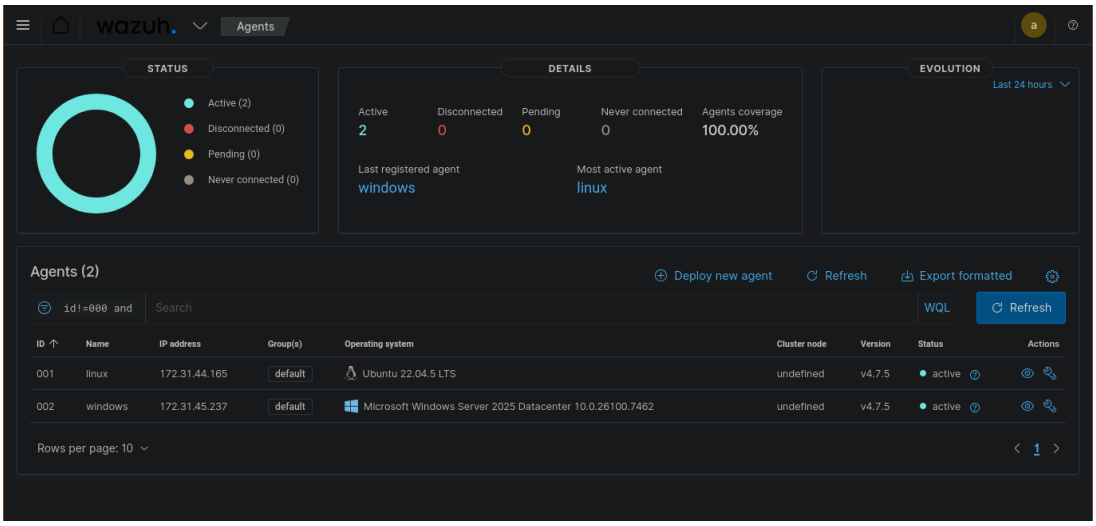

FIGURE 3.2 – Installation du serveur Wazuh

3.0.4 Enrôlement des agents Linux et Windows

Les agents ont été enrôlés via le dashboard Wazuh, garantissant une communication sécurisée avec le serveur.

```
ubuntu@ip-172-31-44-165:~$ ping 172.31.37.232
PING 172.31.37.232 (172.31.37.232) 56(84) bytes of data.
64 bytes from 172.31.37.232: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 172.31.37.232: icmp_seq=2 ttl=64 time=0.213 ms
64 bytes from 172.31.37.232: icmp_seq=3 ttl=64 time=0.229 ms
^C
--- 172.31.37.232 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2048ms
rtt min/avg/max/mdev = 0.213/0.487/1.021/0.377 ms
<ANAGER='172.31.37.232' WAZUH_AGENT_NAME='linux' dpkg -i ./wazuh-agent_4.7.5-1_amd64.deb
--2026-01-05 16:31:07--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.5-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 13.226.209.39, 13.226.209.78, 13.226.209.93, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|13.226.209.39|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9378818 (8.9M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.7.5-1_amd64.deb'

wazuh-agent_4.7.5-1_amd64.deb     100%[=============================================================>]   8.94M  --.-KB/s    in 0.02s

2026-01-05 16:31:07 (358 MB/s) - 'wazuh-agent_4.7.5-1_amd64.deb' saved [9378818/9378818]

Selecting previously unselected package wazuh-agent.
(Reading database ... 65993 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.7.5-1_amd64.deb ...
Unpacking wazuh-agent (4.7.5-1) ...
Setting up wazuh-agent (4.7.5-1) ...
```

FIGURE 3.3 – Agent Linux enregistré dans Wazuh

3.0.5 Scénarios d'attaque et détection par Wazuh

Afin d'illustrer le fonctionnement opérationnel d'une plateforme SIEM/EDR et d'adopter une approche réaliste proche de celle d'un Security Operations Center (SOC), plusieurs scénarios d'attaque ont été volontairement simulés dans le laboratoire.

Ces scénarios correspondent à des techniques largement utilisées par les attaquants dans des environnements réels, aussi bien lors de la phase d'accès initial que lors des phases post-compromission.

Trois scénarios d'attaque principaux ont été mis en œuvre :

— Attaque par bruteforce SSH sur un système Linux,

— Attaque par bruteforce RDP sur un système Windows,
— Création d'un utilisateur local ajouté au groupe Administrateurs (élévation de privilèges).

Scénario 1 : Attaque par bruteforce SSH (Linux)

Description de l'attaque Le protocole SSH constitue l'un des principaux points d'entrée sur les systèmes Linux exposés. Un attaquant peut tenter de deviner les identifiants d'un compte valide en multipliant les tentatives de connexion avec des mots de passe incorrects. Cette technique, appelée **bruteforce SSH**, est fréquemment automatisée et vise à obtenir un accès initial au système.

Dans ce scénario, plusieurs tentatives de connexion SSH échouées ont été générées vers l'instance Linux supervisée.

Objectif de l'attaquant L'objectif principal de l'attaquant est :
— d'identifier l'existence de comptes valides,
— d'obtenir un accès non autorisé au système,
— de préparer une phase ultérieure d'élévation de privilèges ou de mouvement latéral.

Détection par Wazuh Wazuh surveille les journaux d'authentification Linux, notamment le fichier `/var/log/auth.log`. Chaque tentative de connexion SSH échouée génère un événement de type *authentication failure*.

Lorsque plusieurs échecs sont détectés sur une courte période, Wazuh corrèle ces événements et déclenche des alertes de niveau élevé, associées aux techniques MITRE ATT&CK liées à l'accès aux identifiants (*Credential Access*).

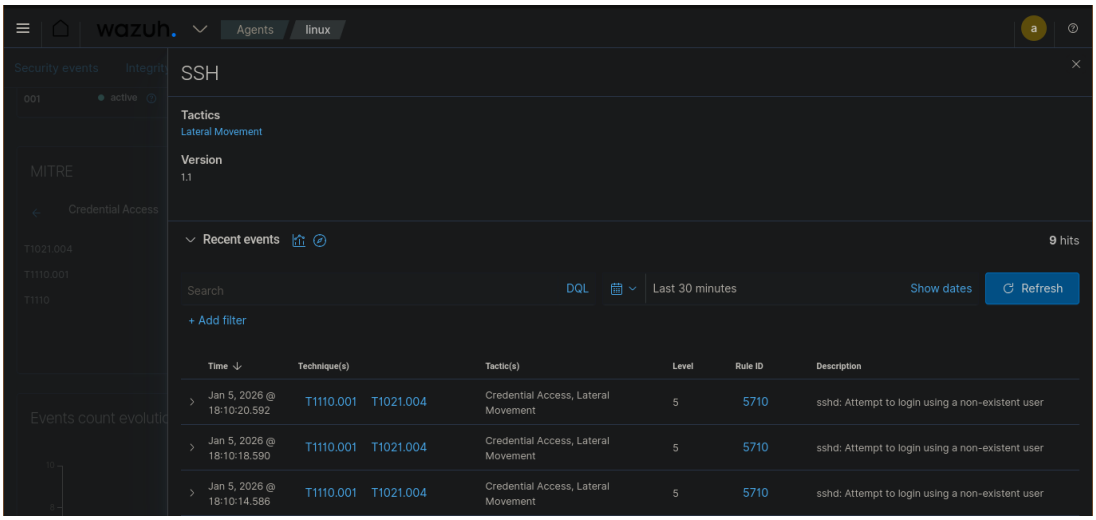

FIGURE 3.4 – Détection par Wazuh des tentatives de bruteforce SSH sur le système Linux

Analyse SOC Ce type d'alerte permet à un analyste SOC d'identifier rapidement une tentative d'attaque automatisée et de mettre en place des contre-mesures telles que le blocage de l'adresse source ou le renforcement des politiques d'authentification.

Scénario 2 : Attaque par bruteforce RDP (Windows)

Description de l'attaque Le protocole RDP est l'un des vecteurs d'attaque les plus exploités dans les environnements Windows. Une attaque par bruteforce RDP consiste à effectuer de multiples tentatives de connexion avec des mots de passe incorrects afin de compromettre un compte utilisateur ou administrateur.

Dans ce scénario, plusieurs connexions RDP ont été tentées avec des identifiants invalides sur le serveur Windows supervisé.

Objectif de l'attaquant Les objectifs de l'attaquant sont :

- obtenir un accès interactif à la machine,
- compromettre un compte à privilèges élevés,
- se positionner durablement dans le système d'information.

Détection par Wazuh Sous Windows, chaque échec de connexion RDP génère un événement de sécurité **Event ID 4625**. Ces événements sont collectés par l'agent Wazuh, puis transmis au serveur pour analyse et corrélation.

Wazuh est capable d'identifier les tentatives répétées de connexion échouée et de générer des alertes indiquant une activité suspecte de type bruteforce.

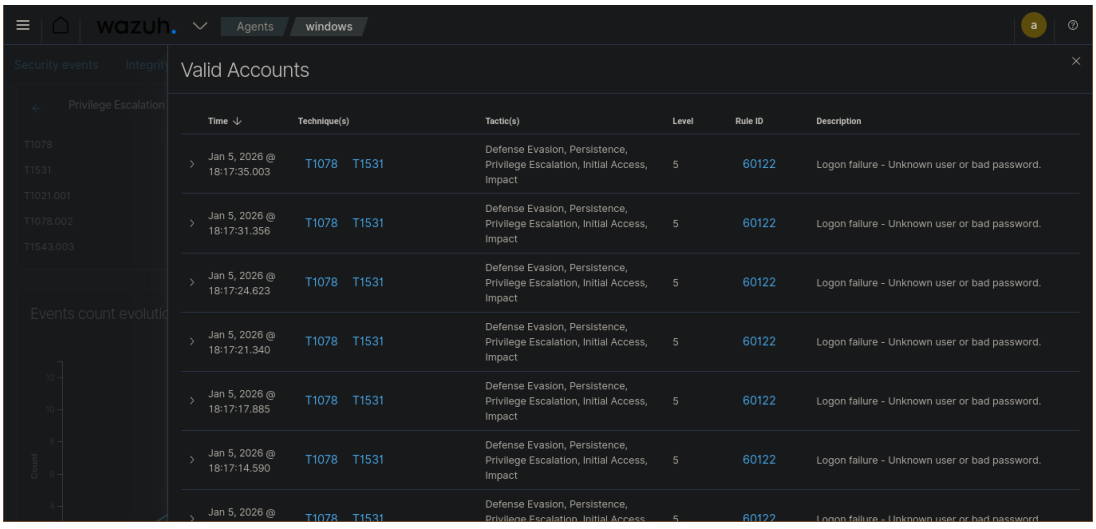

FIGURE 3.5 – Alertes Wazuh liées aux échecs de connexion RDP (Event ID 4625)

Analyse SOC La détection précoce de ce type d'attaque est critique, car les attaques par bruteforce RDP sont souvent utilisées comme point d'entrée initial dans les campagnes de ransomware et d'intrusion ciblée.

Scénario 3 : Création d'un utilisateur administrateur (Élévation de privilèges)

Description de l'attaque Ce scénario représente une phase avancée de l'attaque, dite **post-compromission**. Après avoir obtenu un accès initial, un attaquant cherche généralement à maintenir un accès persistant et à augmenter ses privilèges.

Dans ce cas, un nouvel utilisateur local a été créé sur le système Windows puis ajouté au groupe **Administrators**, ce qui confère des droits complets sur la machine.

Commandes utilisées

```
net user labuser password123! /add
net localgroup administrators labuser /add
```

Objectif de l'attaquant Cette technique permet à l'attaquant :

- d'assurer une persistance durable,
- de contourner les contrôles de sécurité,
- d'exécuter des actions critiques sur le système.

Détection par Wazuh Wazuh surveille les événements Windows liés à la gestion des comptes et des groupes, notamment les événements de création d'utilisateur et d'ajout à des groupes privilégiés (Event ID 4720 et 4732).

Ces événements sont considérés comme critiques car ils indiquent une élévation de privilèges potentiellement malveillante.

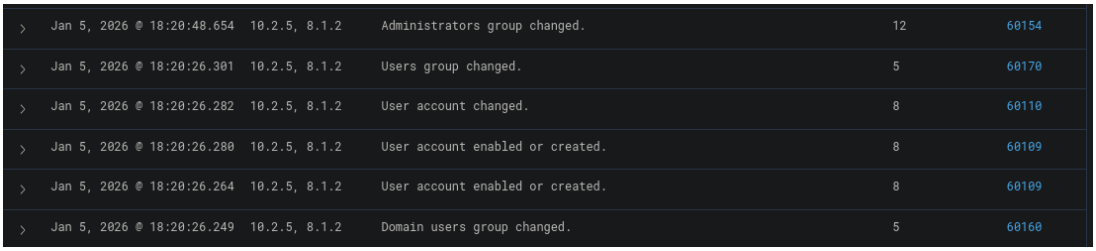

FIGURE 3.6 – Détection par Wazuh de la création d'un utilisateur et de son ajout au groupe Administrators

Analyse SOC Pour un SOC, ce scénario constitue un indicateur fort de compromission avancée. Une réponse immédiate est nécessaire afin de contenir l'incident, investiguer l'origine de l'attaque et restaurer l'intégrité du système.

Synthèse des scénarios d'attaque

Les scénarios présentés couvrent plusieurs phases clés d'une attaque réelle :

- Accès initial (bruteforce SSH et RDP),
- Tentatives répétées d'authentification,
- Élévation de privilèges et persistance.

Ils démontrent l'efficacité de Wazuh en tant que solution SIEM/EDR capable de détecter, corréler et contextualiser des événements de sécurité dans un environnement Cloud multi-systèmes.

Chapitre 4

Conclusion

Ce projet a permis de mettre en œuvre une plateforme SIEM/EDR complète dans un environnement Cloud réel. Les scénarios testés démontrent l'efficacité de Wazuh pour la supervision multi-OS, la détection d'incidents et l'analyse de comportements malveillants.

Cette approche constitue une base solide pour la compréhension du fonctionnement d'un SOC moderne et des enjeux liés à la sécurité des endpoints.