

**Département Mathématique et Informatique**

**Filière :**  
**«Ingénierie Informatique,**  
**Cybersécurité et confiance numérique»**

**Mise en place d'une solution de  
sécurité des endpoints et de  
supervision SIEM avec Wazuh dans  
un environnement AWS**

2025/2026

Réalisé par :

Aya OUZARF

Encadré par :

Pr. Azeddine KHIAT

## REMERCIEMENTS

Au terme de ce projet, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à sa réalisation.

Mes premiers remerciements vont au **Professeur Azeddine KHIAT** pour son encadrement rigoureux, ses conseils avisés et sa disponibilité tout au long de ce projet. Son expertise et ses orientations ont été déterminantes pour mener à bien ce travail.

Je remercie sincèrement l'**ensemble du corps professoral** du Département Mathématique et Informatique pour la qualité de l'enseignement dispensé, qui a constitué un socle solide pour ce projet.

Mes remerciements vont également à l'**établissement** pour avoir mis à disposition les ressources nécessaires, notamment l'accès à **AWS Academy** et au **Learner Lab**, infrastructures essentielles pour déployer un environnement Cloud professionnel.

J'exprime ma reconnaissance envers la **communauté open-source Wazuh** et les contributeurs de **Sysmon**, dont les solutions et la documentation ont grandement facilité la mise en œuvre technique.

Je remercie les **membres du jury** pour l'intérêt porté à ce travail et le temps consacré à son évaluation.

Mes remerciements vont aussi à **mes camarades de promotion** pour les échanges enrichissants et le soutien mutuel durant ces années d'études.

À tous, merci infiniment.

# TABLE DES MATIÈRES

<b>REMERCIEMENTS -----</b>	<b>2</b>
<b>TABLE DES MATIÈRES -----</b>	<b>3</b>
<b>Tables des figures-----</b>	<b>5</b>
<b>PARTIE I : CONTEXTE ET FONDAMENTAUX-----</b>	<b>6</b>
<b>    1. INTRODUCTION GÉNÉRALE -----</b>	<b>6</b>
1.2 Contexte du projet -----	6
1.2 Lien GitHub -----	6
<b>    2. PROBLÉMATIQUES ET OBJECTIFS-----</b>	<b>7</b>
2.1 Problématiques identifiées -----	7
2.2 Objectifs du projet-----	7
<b>PARTIE II : CADRE THÉORIQUE -----</b>	<b>8</b>
<b>    1. CONCEPTS FONDAMENTAUX -----</b>	<b>8</b>
1.1 SIEM (Security Information and Event Management) -----	8
1.2 EDR (Endpoint Detection and Response)-----	9
1.3 Cloud Security avec AWS -----	10
<b>    2. TECHNOLOGIES UTILISÉES -----</b>	<b>11</b>
2.1 Wazuh-----	11
2.2 Sysmon -----	13
<b>PARTIE III : ARCHITECTURE ET DÉPLOIEMENT -----</b>	<b>14</b>
<b>    1. ARCHITECTURE TECHNIQUE -----</b>	<b>14</b>
1.1 Vue d'ensemble -----	14
1.2 Spécifications des instances -----	14
1.3 Configuration réseau -----	15
<b>    2. DÉPLOIEMENT DE L'INFRASTRUCTURE AWS -----</b>	<b>16</b>
2.1 Création des instances EC2-----	16
2.2 Configuration post-déploiement -----	17
<b>    3. INSTALLATION WAZUH SERVER -----</b>	<b>18</b>
3.1 Connexion SSH et préparation -----	18
3.2 Installation Wazuh All-in-One -----	18
3.3 Vérification des services -----	19
3.4 Accès au Dashboard-----	20
<b>    4. ENRÔLEMENT DES AGENTS -----</b>	<b>22</b>
4.1 Agent Linux -----	22
4.2 Agent Windows + Sysmon-----	24
<b>PARTIE IV : TESTS ET ANALYSES -----</b>	<b>27</b>

<b>1. SCÉNARIOS DE SÉCURITÉ TESTÉS-----</b>	<b>27</b>
1.1 SSH Bruteforce (Linux)-----	27
1.2 Élévation de priviléges et FIM (Linux) -----	29
1.3 Échecs de connexion RDP (Windows) -----	32
1.4 Création d'utilisateur Windows -----	34
1.5 Événements Sysmon (EDR Windows) -----	36
<b>2. ANALYSE COMPARATIVE SIEM VS EDR -----</b>	<b>42</b>
2.1 Différences fondamentales -----	42
2.2 Complémentarité démontrée -----	43
2.3 Cas d'usage optimal pour chaque approche-----	43
<b>3.IAM/PAM ET GESTION DES ACCÈS-----</b>	<b>43</b>
3.1 Identity and Access Management (IAM) -----	43
3.2 Privileged Access Management (PAM)-----	44
3.3 Principes de sécurité IAM/PAM-----	44
<b>4. THREAT HUNTING ET DÉTECTION AVANCÉE -----</b>	<b>44</b>
4.1 Concept de Threat Hunting -----	44
4.2 Requêtes de Threat Hunting -----	45
4.3 MITRE ATT&CK Mapping-----	46
<b>PARTIE V : RÉSULTATS ET CONCLUSION-----</b>	<b>47</b>
<b>1. RÉSULTATS OBTENUS-----</b>	<b>47</b>
<b>2. CONCLUSION GÉNÉRALE-----</b>	<b>48</b>

## Tables des figures

Figure 1 : schématique de l'architecture du lab-----	14
Figure 2 : Configuration SG -----	16
Figure 3: tableau de bord AWS EC2-----	17
Figure 4 : Installation Wazuh -----	19
Figure 5: Vérification des services Wazuh-----	20
Figure 6 : Page de connexion Wazuh -----	21
Figure 7 : Interface du tableau de bord Wazuh-----	21
Figure 8 : Interface "Deploy new agent" -----	22
Figure 9 : Installation de l'agent Wazuh sur le Linux-Client-----	23
Figure 10 : Linux-Clien "Active" -----	23
Figure 11 : Installation de l'agent Wazuh sur le Windows-Client-----	25
Figure 12 : l'agent Windows-Client "Active"-----	25
Figure 13 : Installation et configuration de Sysmon-----	26
Figure 14 : Tentatives SSH échouées-----	27
Figure 15 : Les alerte SSH brute force -----	28
Figure 16 : Détails d'une alerte SSH brute force -----	28
Figure 17 : Exécution de la commande sudo su -----	29
Figure 18 : Alerte sudo-----	30
Figure 19 : Modification /etc/passwd-----	31
Figure 20 : Alert File Integrity Monitoring (FIM)-----	31
Figure 21 : Détails complets de l'alerte FIM-----	32
Figure 22 : Tentatives de connexion RDP échouées -----	33
Figure 23 : Alert de connexion RDP échouées -----	33
Figure 24 : Detaille d' alerte connexion RDP échouées-----	34
Figure 25 : Création d'un utilisateur local "labuser" -----	35
Figure 26 : Ajout l'utilisateur au groupe Administrateurs -----	35
Figure 27 : Alert l'ajout d'un utilisateur -----	36
Figure 28 : Service Sysmon64 actif-----	37
Figure 29 : Collection des événements Sysmon-----	37
Figure 30 : L'Event ID 1 -----	38
Figure 31 : Alert d'Event ID 1 -----	38
Figure 32 : L'Event ID 11 -----	39
Figure 33 : Alert d'Event ID 11-----	39
Figure 34 : L'Event ID 3 -----	40
Figure 35 : Alert d'Event ID 3 -----	40
Figure 36 : Activité PowerShell suspecte -----	41
Figure 37 : Alert d'activité PowerShell suspecte -----	41

# PARTIE I : CONTEXTE ET FONDAMENTAUX

## 1. INTRODUCTION GÉNÉRALE

Dans le contexte actuel de transformation digitale, la cybersécurité est devenue une priorité stratégique. Les statistiques sont alarmantes : selon l'ANSSI, les cyberattaques ont augmenté de 400% entre 2019 et 2024. Le coût moyen d'une violation de données atteint 4,45 millions de dollars (IBM Security Report 2024).

Ce rapport présente la mise en œuvre d'une plateforme complète de supervision basée sur **Wazuh**, déployée sur **AWS Cloud**. L'objectif est de démontrer comment combiner les approches **SIEM** (Security Information and Event Management) et **EDR** (Endpoint Detection and Response) pour créer une solution robuste capable de détecter et analyser les menaces sur des environnements Linux et Windows.

### 1.2 Contexte du projet

Ce projet s'inscrit dans le cadre de la formation en cybersécurité et reproduit un environnement de **Security Operations Center (SOC)** moderne. Les compétences développées couvrent :

- Déploiement d'infrastructure Cloud (AWS)
- Configuration de solutions SIEM
- Analyse d'événements de sécurité
- Investigation d'incidents
- Threat hunting

### 1.2 Lien GitHub

L'ensemble du projet est documenté sur GitHub :

 <https://github.com/AyaOuzarf/wazuh-siem-edr-lab>

Le dépôt contient :

- Fichiers de configuration (Wazuh, Sysmon)
- Captures d'écran détaillées
- Scripts d'installation
- Documentation complète

## 2. PROBLÉMATIQUES ET OBJECTIFS

### 2.1 Problématiques identifiées

Les organisations modernes rencontrent plusieurs défis en cybersécurité :

- **Manque de visibilité** : les solutions sont souvent déployées en silos, rendant difficile une vue globale des événements.
- **Temps de détection élevé** : les intrusions peuvent passer inaperçues en moyenne 207 jours, laissant les attaquants agir longtemps.
- **Environnements hétérogènes** : la diversité des systèmes (Windows, Linux, Cloud, conteneurs) complique l'analyse des logs.
- **Volume de données important** : des téraoctets de logs sont générés quotidiennement, rendant l'analyse manuelle impossible.
- **Attaques sophistiquées** : les méthodes avancées (malwares sans fichiers, exploitation des outils existants) échappent aux solutions traditionnelles.

### 2.2 Objectifs du projet

**Objectif général** Mettre en œuvre une plateforme de supervision complète, combinant SIEM et EDR, sur AWS Cloud, capable de détecter et analyser les menaces sur Linux et Windows.

#### Objectifs spécifiques

##### 1. Infrastructure Cloud sécurisée

- Créer un VPC AWS avec isolation réseau
- Configurer des Security Groups (principe du moindre privilège)
- Déployer 3 instances EC2
- Documenter l'architecture

##### 2. SIEM avec Wazuh

- Installer Wazuh All-in-One (Manager + Indexer + Dashboard)
- Configurer la collecte multi-sources
- Créer des règles de corrélation
- Mettre en place des dashboards

##### 3. Capacités EDR

- Déployer agents Wazuh (Linux + Windows)
- Intégrer Sysmon pour visibilité avancée
- Configurer File Integrity Monitoring
- Activer détection comportementale

#### 4. Tests et validation

- Simuler attaques (bruteforce, élévation privilèges)
- Vérifier détection en temps réel
- Analyser alertes générées
- Valider corrélation multi-sources

#### 5. Analyse

- Comparer SIEM vs EDR
- Analyser IAM/PAM
- Réaliser du threat hunting
- Proposer optimisations

## PARTIE II : CADRE THÉORIQUE

### 1. CONCEPTS FONDAMENTAUX

#### 1.1 SIEM (Security Information and Event Management)



**Définition :** Un SIEM combine deux approches historiques :

- **SIM** (Security Information Management) : Collecte et stockage centralisé des logs
- **SEM** (Security Event Management) : Analyse en temps réel et corrélation

Le terme SIEM a été popularisé par Gartner en 2005 pour désigner la convergence de ces approches.

**Fonctionnement :** Le SIEM fonctionne en 6 étapes :

1. **Collecte** : Réception de logs depuis multiples sources (OS, applications, réseau, sécurité, Cloud)
2. **Normalisation** : Transformation des logs en format unifié
3. **Enrichissement** : Ajout de contexte (géolocalisation, réputation, threat intelligence)
4. **Corrélation** : Identification de patterns d'attaque
5. **Alerting** : Notification selon la sévérité (Low, Medium, High, Critical)
6. **Visualisation** : Dashboards et reporting

### Cas d'usage principaux

- Détection d'intrusion (bruteforce, exploitation, exfiltration)
- Monitoring des accès privilégiés
- Détection d'anomalies comportementales
- Conformité réglementaire (RGPD, PCI-DSS, ISO 27001)

### Limites

- Complexité de configuration
- Coût élevé des solutions commerciales
- Problème des faux positifs

## 1.2 EDR (Endpoint Detection and Response)

**Définition** Un EDR surveille en continu les endpoints (postes, serveurs) pour détecter et répondre aux menaces avancées.

### Évolution

- **Génération 1** (1990s) : Antivirus à signatures
- **Génération 2** (2010s) : Next-Gen Antivirus (NGAV) avec ML
- **Génération 3** (2015+) : EDR avec Response et investigation

- **Génération 4 (2020+)** : XDR (Extended Detection Response)

### Fonctionnalités clés

- **Monitoring continu** : Processus, connexions réseau, fichiers, registre
- **Détection comportementale** : Analyse des chaînes d'exécution
- **Threat Intelligence** : Hash malveillants, IOCs, YARA rules
- **Investigation forensique** : Timeline complète
- **Response automatisée** : Isolation, kill processus

### EDR vs Antivirus

Critère	Antivirus	EDR
Approche	Signatures	Comportemental
Couverture	Fichiers	Processus, réseau, registre
Détection	Known threats	Unknown threats
Investigation	Limitée	Timeline complète

### 1.3 Cloud Security avec AWS



**Amazon Web Services (AWS)** est le leader mondial du Cloud computing, proposant plus de 200 services. AWS opère selon un **modèle de responsabilité partagée** :

#### Responsabilité AWS (Sécurité DU Cloud)

- Infrastructure physique des datacenters
- Réseau global
- Hyperviseur de virtualisation

- Services managés

### **Responsabilité Client (Sécurité DANS le Cloud)**

- Configuration des Security Groups
- Gestion IAM (identités et accès)
- Chiffrement des données
- Mises à jour OS et applications
- Monitoring et logging

### **Services AWS utilisés dans ce projet**

**VPC (Virtual Private Cloud)** Réseau isolé logiquement dans le Cloud AWS. Permet de contrôler complètement l'environnement réseau virtuel avec subnets publics/privés.

**EC2 (Elastic Compute Cloud)** Instances virtuelles scalables. Différents types d'instances (t2, t3) adaptés aux besoins (CPU, RAM).

**Security Groups** Pare-feu virtuel au niveau instance. Règles stateful contrôlant le trafic inbound/outbound.

### **Avantages du Cloud**

- Scalabilité et élasticité
- Haute disponibilité (99.99%)
- Pay-as-you-go (paiement à l'usage)
- Déploiement rapide

## **2.TECHNOLOGIES UTILISÉES**

### **2.1 Wazuh**

# wazuh.

**Présentation** Wazuh est une plateforme de sécurité open-source (licence GPLv3) qui fournit des capacités SIEM et XDR. Basée sur OSSEC HIDS et intégrant Opensearch.

### **Composants principaux**

ENSET, Avenue Hassan II - B.P. 159 - Mohammedia - Maroc

☎ 05 23 32 22 20 / 05 23 32 35 30 – Fax : 05 23 32 25 46 - Site Web: [www.enset-media.ac.ma](http://www.enset-media.ac.ma)

E-Mail : [enset-media@enset-media.ac.ma](mailto:enset-media@enset-media.ac.ma)

## Wazuh Manager

- Collecte événements des agents
- Applique règles de détection
- Corrèle événements multi-sources
- Stocke alertes dans l'Indexer

## Wazuh Indexer (OpenSearch)

- Indexe événements pour recherche rapide
- Stockage à long terme
- Réplication en cluster

## Wazuh Dashboard

- Interface Web de visualisation
- Dashboards préconfigurés
- Requêtes avancées

## Wazuh Agent

- Déployé sur endpoints
- Collecte logs système/applicatifs
- Exécute scans de sécurité (FIM, SCA)
- Communique via port 1514/TCP

## Modules Wazuh

- **Log Analysis** : Analyse avec règles
- **FIM** : File Integrity Monitoring
- **SCA** : Security Configuration Assessment
- **Vulnerability Detection** : Détection CVE
- **Threat Intelligence** : Intégration IOCs

## 2.2 Sysmon

**Présentation** Sysmon (System Monitor) est un service Windows développé par Microsoft Sysinternals qui enregistre les activités système dans le journal d'événements.

### Pourquoi Sysmon ?

- Visibilité avancée sur processus, réseau, fichiers
- Détection de comportements suspects
- Gratuit et Microsoft officiel
- Logs structurés pour Wazuh

### Event IDs Sysmon essentiels

Event ID	Description	Usage Cybersécurité
1	Process Creation	Malware detection, chaîne exécution
3	Network Connection	C2 communication, exfiltration
5	Process Terminated	Analyse durée de vie
7	Image Loaded	DLL injection
8	CreateRemoteThread	Code injection
10	ProcessAccess	Credential dumping
11	FileCreate	Ransomware, dropper
22	DNS Query	Domaines malveillants

**Intégration Wazuh + Sysmon** Sysmon écrit dans "Microsoft-Windows-Sysmon/Operational"  
 → Wazuh Agent collecte ce canal → Manager applique règles de détection.

## PARTIE III : ARCHITECTURE ET DÉPLOIEMENT

### 1.ARCHITECTURE TECHNIQUE

#### 1.1Vue d'ensemble

L'architecture déployée comprend 3 instances EC2 dans un VPC AWS, reproduisant un environnement SOC avec serveur centralisé et endpoints supervisés.

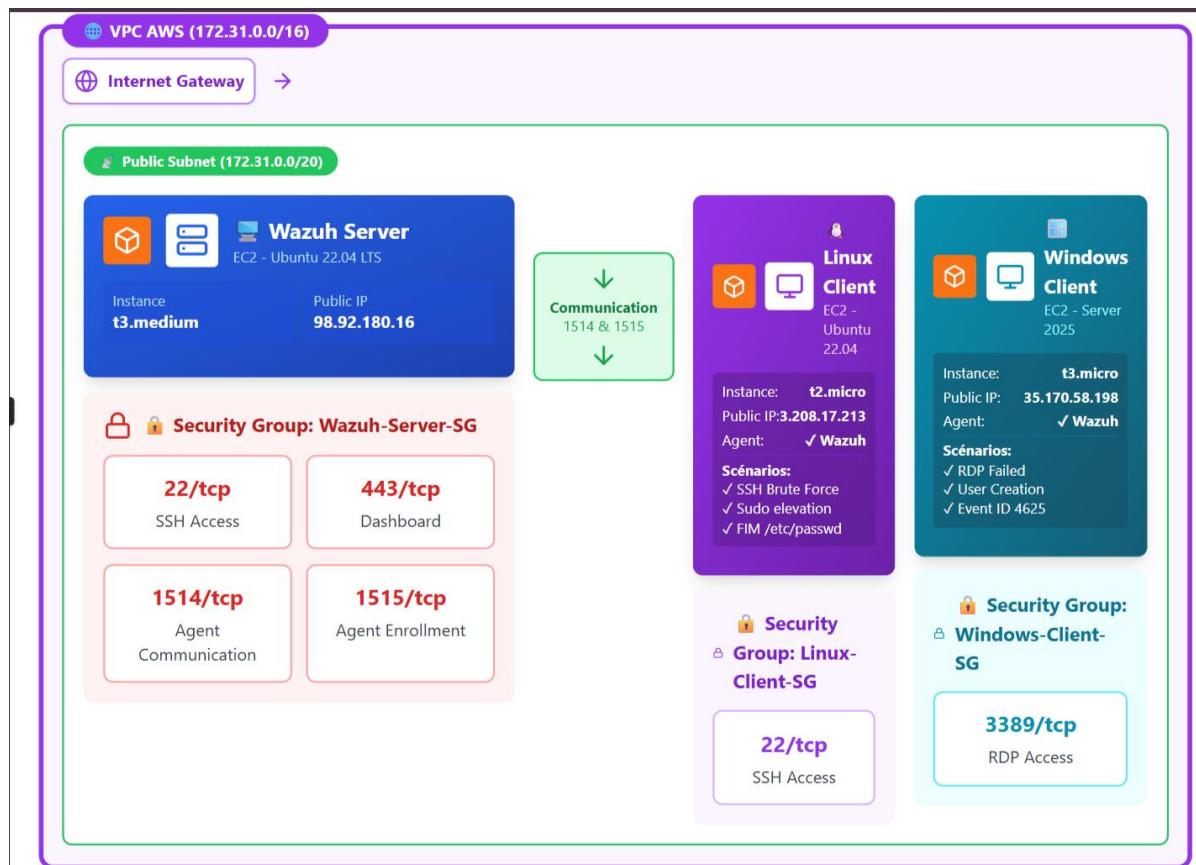


Figure 1 : schématique de l'architecture du lab

Vue schématique de l'architecture du lab montrant les trois instances EC2 (Wazuh-Server, Linux-Client, Windows-Client) dans un VPC AWS, avec les flux réseau principaux et les règles de sécurité.

#### 1.2Spécifications des instances

Instance	OS	Type	vCPU	RAM	Storage	IP Publique	Rôle
Wazuh-Server	Ubuntu 22.04	t3. medium	2	8 GB	30 GB SSD	18.213.113.165	Manager+Indexer+Dashboard

Instance	OS	Type	vCPU	RAM	Storage	IP Publique	Rôle
Linux-Client	Ubuntu 22.04	t2.micro	1	1 GB	8 GB SSD	98.91.206.16	Endpoint Linux supervisé
Windows-Client	Win Server 2025	t2. micro	2	4 GB	30 GB SSD	MyIP	Endpoint Windows supervisé

### Justification des choix d'instances

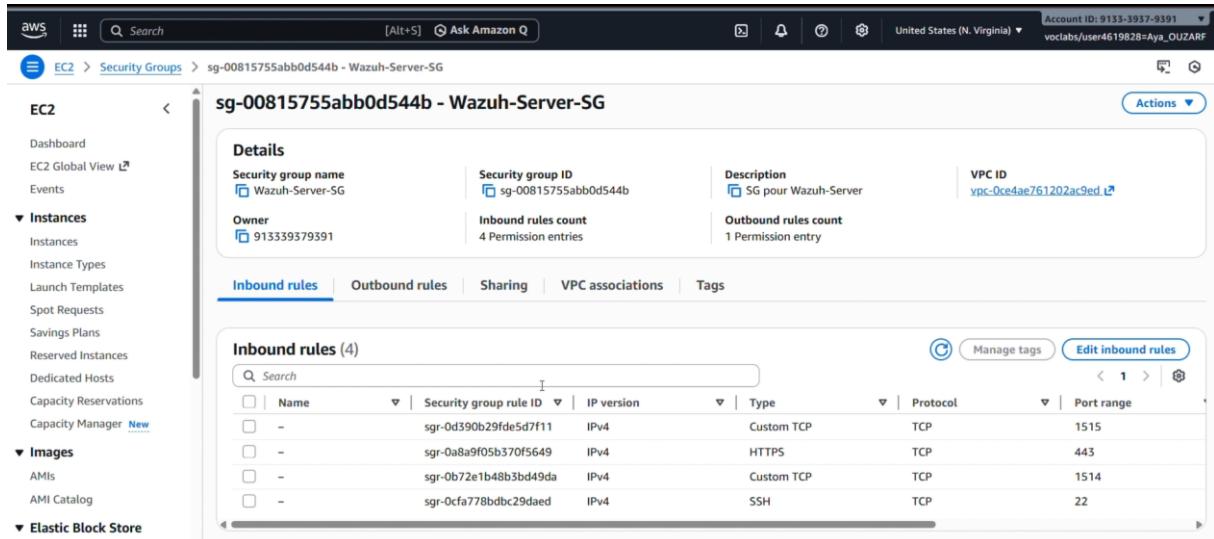
- t3.medium pour Wazuh-Server** : L'Indexer Opensearch nécessite minimum 4 GB de RAM pour fonctionner correctement. Avec 8 GB, nous disposons d'une marge confortable pour l'indexation et les requêtes simultanées.
- t2.micro pour Linux-Client** : L'agent Wazuh est extrêmement léger (overhead < 100 MB RAM, < 5% CPU). Une instance t2.micro suffit largement pour la démonstration.
- t3.micro pour Windows-Client** : Windows Server et Sysmon nécessitent plus de ressources que l'agent Linux. Les 4 GB de RAM permettent un fonctionnement fluide de l'OS et des outils de sécurité.

### 1.3 Configuration réseau

**VPC et Subnet** : Toutes les instances sont déployées dans le même VPC et le même subnet pour simplifier la communication. En production, une architecture multi-tier avec subnets publics/privés et bastion hosts serait recommandée.

**Security Group : SG-Wazuh-Lab** : Le Security Group définit les règles de pare-feu au niveau de chaque instance. Configuration appliquée :

Port	Protocole	Source	Usage
22	TCP	Mon IP	SSH administration serveurs
443	TCP	Mon IP	Dashboard Wazuh HTTPS
1514	TCP	Security Group	Communication agents → Manager
1515	TCP	Security Group	Enrollment automatique agents
3389	TCP	Mon IP	RDP Windows-Client



Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0d390b29fde5d7f11	IPv4	Custom TCP	TCP	1515
-	sgr-0a8a9f05b370f5649	IPv4	HTTPS	TCP	443
-	sgr-0b72e1b48b3bd49da	IPv4	Custom TCP	TCP	1514
-	sgr-0cfa778bdbc29daed	IPv4	SSH	TCP	22

Figure 2 : Configuration SG

Configuration détaillée des règles Inbound du Security Group 'SG-Wazuh-Lab' avec les ports autorisés (22, 443, 1514, 1515, 3389) et leurs sources respectives.

### Principe de sécurité appliqué : Moindre privilège

- Les ports administratifs (22, 3389, 443) sont limités à l'IP publique de l'administrateur uniquement
- Les ports de communication inter-instances (1514, 1515) n'acceptent que le trafic provenant du même Security Group
- Tout le reste du trafic est implicitement bloqué (deny all non explicitement autorisé)

## 2. DÉPLOIEMENT DE L'INFRASTRUCTURE AWS

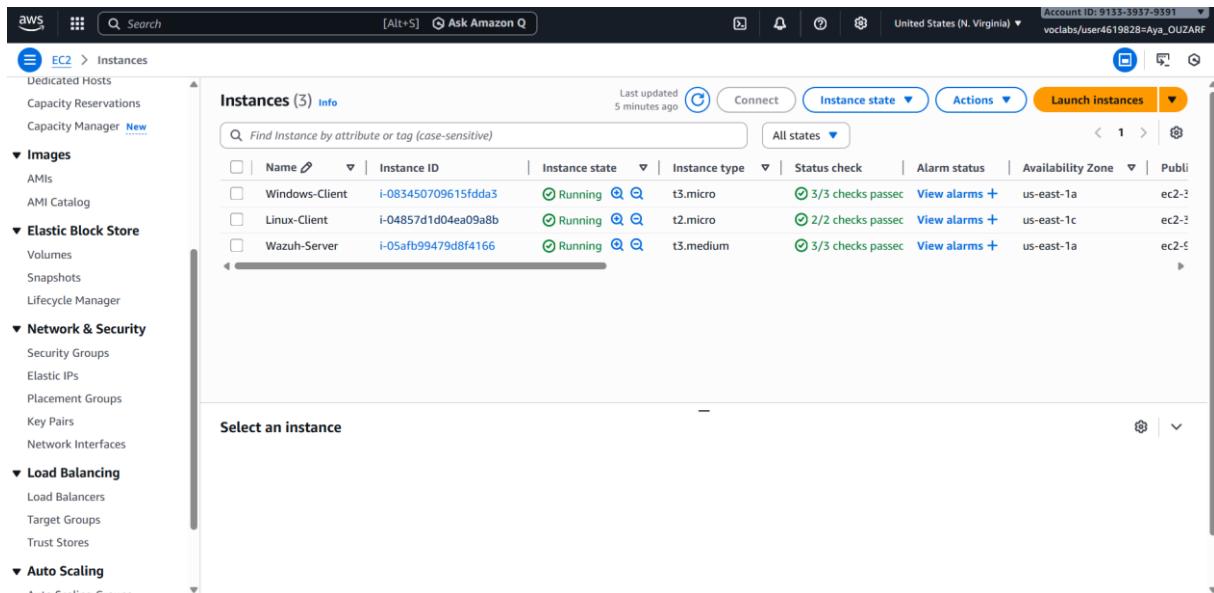
### 2.1 Crédation des instances EC2

La première étape consiste à déployer les 3 instances EC2 sur AWS Learner Lab. Après connexion à AWS Academy et démarrage du lab, nous accédons à la console EC2.

#### Processus de création

- Sélection des AMI (Amazon Machine Images) :
  - Ubuntu Server 22.04 LTS pour Wazuh-Server et Linux-Client
  - Windows Server 2025 Datacenter pour Windows-Client

2. Choix des types d'instances selon les spécifications définies (t3.medium, t2.micro, t3.micro)
3. Configuration réseau : Toutes les instances dans le même VPC, même subnet, avec IP publiques auto-assignées
4. Création et téléchargement de la paire de clés SSH (format .pem) pour l'authentification sécurisée
5. Application du Security Group SG-Wazuh-Lab à toutes les instances
6. Dimensionnement du stockage (30 GB pour Wazuh-Server et Windows-Client, 8 GB pour Linux-Client)



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Windows-Client	i-083450709615fdda3	Running	t3.micro	3/3 checks passed	<a href="#">View alarms</a>	us-east-1a	ec2-2
Linux-Client	i-04857d1d04ea09a8b	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-3
Wazuh-Server	i-05afb99479d8f4166	Running	t3.medium	3/3 checks passed	<a href="#">View alarms</a>	us-east-1a	ec2-5

Figure 3: tableau de bord AWS EC2

Vue du tableau de bord AWS EC2 confirmant le déploiement réussi des trois instances (Wazuh-Server, Linux-Client, Windows-Client) avec leurs états "Running", adresses IP publiques et types d'instances.

## 2.2 Configuration post-déploiement

### Attribution de noms explicites

Les instances sont nommées de manière claire pour faciliter l'identification :

- Wazuh-Server pour le serveur central
- Linux-Client pour l'endpoint Ubuntu

- Windows-Client pour l'endpoint Windows

### Vérification de la connectivité

Avant de procéder aux installations, nous vérifions que :

- Les instances sont bien à l'état "Running"
- Les Status Checks (système et instance) sont réussis
- Les IP publiques sont correctement assignées
- Le Security Group est appliqué et les règles sont actives

## 3. INSTALLATION WAZUH SERVER

### 3.1 Connexion SSH et préparation

La connexion au serveur Ubuntu s'effectue via SSH en utilisant la clé privée téléchargée lors de la création de l'instance.

```
ssh -i wazuh-key.pem ubuntu@18.213.113.165
```

Une fois connecté, la mise à jour complète du système est essentielle pour éviter les conflits de dépendances et les failles de sécurité connues.

```
sudo apt update && sudo apt -y upgrade
```

Cette étape peut prendre plusieurs minutes selon le nombre de packages à mettre à jour.

### 3.2 Installation Wazuh All-in-One

Wazuh fournit un script d'installation automatisé qui simplifie grandement le déploiement en installant et configurant tous les composants (Manager, Indexer, Dashboard) en une seule commande.

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
```

```
sudo bash wazuh-install.sh -a
```

```
ubuntu@ip-172-31-7-50:~$ sudo bash wazuh-install.sh -a -i
01/01/2026 21:12:52 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.5
01/01/2026 21:12:52 INFO: Verbose logging redirected to /var/log/wazuh-install.log
01/01/2026 21:13:00 WARNING: Hardware and system checks ignored.
01/01/2026 21:13:00 INFO: Wazuh web interface port will be 443.
01/01/2026 21:13:06 INFO: --- Dependencies ---
01/01/2026 21:13:06 INFO: Installing apt-transport-https.
01/01/2026 21:13:11 INFO: Wazuh repository added.
01/01/2026 21:13:11 INFO: --- Configuration files ---
01/01/2026 21:13:14 INFO: Generating configuration files.
01/01/2026 21:13:14 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
01/01/2026 21:13:14 INFO: --- Wazuh indexer ---
01/01/2026 21:13:14 INFO: Starting Wazuh indexer installation.
01/01/2026 21:14:15 INFO: Wazuh indexer installation finished.
01/01/2026 21:14:15 INFO: Wazuh indexer post-install configuration finished.
01/01/2026 21:14:19 INFO: Starting service wazuh-indexer.
01/01/2026 21:14:30 INFO: wazuh-indexer service started.
01/01/2026 21:14:30 INFO: Initializing Wazuh indexer cluster security settings.
01/01/2026 21:14:47 INFO: Wazuh indexer cluster initialized.
01/01/2026 21:14:47 INFO: --- Wazuh server ---
01/01/2026 21:14:47 INFO: Starting the Wazuh manager installation.
01/01/2026 21:15:43 INFO: Wazuh manager installation finished.
01/01/2026 21:15:43 INFO: Starting service wazuh-manager.
01/01/2026 21:15:55 INFO: wazuh-manager service started.
01/01/2026 21:15:55 INFO: Starting Filebeat installation.
01/01/2026 21:16:06 INFO: Filebeat installation finished.
01/01/2026 21:16:07 INFO: Filebeat post-install configuration finished.
01/01/2026 21:16:07 INFO: Starting service filebeat.
01/01/2026 21:16:08 INFO: filebeat service started.
01/01/2026 21:16:08 INFO: --- Wazuh dashboard ---
01/01/2026 21:16:08 INFO: Starting Wazuh dashboard installation.
01/01/2026 21:16:54 INFO: Wazuh dashboard installation finished.
01/01/2026 21:16:54 INFO: Wazuh dashboard post-install configuration finished.
01/01/2026 21:16:54 INFO: Starting service wazuh-dashboard.
01/01/2026 21:16:55 INFO: wazuh-dashboard service started.
01/01/2026 21:17:30 INFO: Initializing Wazuh dashboard web application.
01/01/2026 21:17:31 INFO: Wazuh dashboard web application initialized.
01/01/2026 21:17:31 INFO: Summary ---
01/01/2026 21:17:31 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
      User: admin
      Password: 8yFzIYsKPE5mn.DL*VyaqUp00sZDLlex
01/01/2026 21:17:31 INFO: Installation finished.
ubuntu@ip-172-31-7-50:~$ |
```

Figure 4 : Installation Wazuh

Exécution du script d'installation Wazuh All-in-One sur le serveur Ubuntu, affichant le résumé final avec l'URL du dashboard, l'utilisateur admin et le mot de passe généré.

### 3.3 Vérification des services

Nous vérifions que les 3 services Wazuh sont actifs.

```
sudo systemctl status wazuh-manager
sudo systemctl status wazuh-indexer
sudo systemctl status wazuh-dashboard
```

```
ubuntu@ip-172-31-7-50:~$ sudo systemctl status wazuh-indexer
● wazuh-indexer.service - Wazuh-indexer
   Loaded: loaded (/usr/lib/systemd/system/wazuh-indexer.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-01-01 21:14:36 UTC; 15min ago
     Docs: https://documentation.wazuh.com
Main PID: 15068 (java)
  Tasks: 74 (limit: 4515)
    Memory: 2.2G (peak: 2.2G)
      CPU: 1min 14.91s
     CGroup: /system.slice/wazuh-indexer.service
             └─15068 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkaddress.cache.ttl=60 -Dopensearch.networkaddress.cache.negativeTTL=60

Jan 01 21:14:16 ip-172-31-7-50 systemd[1]: Starting wazuh-indexer.service - Wazuh-indexer...
Jan 01 21:14:19 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: A terminally deprecated method in java.lang.System has been called
Jan 01 21:14:19 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.OpenSearch (file:/u
Jan 01 21:14:19 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.OpenSearch
Jan 01 21:14:19 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: System::setSecurityManager will be removed in a future release
Jan 01 21:14:21 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: A terminally deprecated method in java.lang.System has been called
Jan 01 21:14:21 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.Security (file:/u
Jan 01 21:14:21 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.Security
Jan 01 21:14:21 ip-172-31-7-50 systemd-entrypoint[15068]: WARNING: System::setSecurityManager will be removed in a future release
Jan 01 21:14:36 ip-172-31-7-50 systemd[1]: Started wazuh-indexer.service - Wazuh-indexer.
ubuntu@ip-172-31-7-50:~$ sudo systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
   Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-01-01 21:17:17 UTC; 13min ago
Main PID: 60886 (node)
  Tasks: 11 (limit: 4515)
    Memory: 218.6M (peak: 239.6M)
      CPU: 13.422s
     CGroup: /system.slice/wazuh-dashboard.service
             └─60886 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-rejections=warn /usr/share/wazuh-dashb

Jan 01 21:29:58 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:58Z", "tags": [], "pid": 60886, "method": "get", "s
Jan 01 21:29:58 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:58Z", "tags": [], "pid": 60886, "method": "post", "s
Jan 01 21:29:58 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:58Z", "tags": [], "pid": 60886, "method": "get", "s
Jan 01 21:29:58 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:58Z", "tags": [], "pid": 60886, "method": "get", "s
Jan 01 21:29:58 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:58Z", "tags": [], "pid": 60886, "method": "get", "s
Jan 01 21:29:59 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:59Z", "tags": [], "pid": 60886, "method": "get", "s
Jan 01 21:29:59 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:59Z", "tags": [], "pid": 60886, "method": "post", "s
Jan 01 21:29:59 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:59Z", "tags": [], "pid": 60886, "method": "post", "s
Jan 01 21:29:59 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "response", "@timestamp": "2026-01-01T21:29:59Z", "tags": [], "pid": 60886, "method": "post", "s
Jan 01 21:30:01 ip-172-31-7-50 opensearch-dashboards[60886]: {"type": "Log", "@timestamp": "2026-01-01T21:30:01Z", "tags": ["error", "opensearch", "data"], "pid": 6
```

*Figure 5: Vérification des services Wazuh*

Vérification des services Wazuh (Manager, Indexer, Dashboard) via systemctl, confirmant leur statut "active (running)" et leur activation au démarrage.

### **3.4 Accès au Dashboard**

Depuis un navigateur, nous accédons à <https://18.213.113.165>. Le certificat SSL auto-signé déclenche un avertissement.

la page de login Wazuh s'affiche.

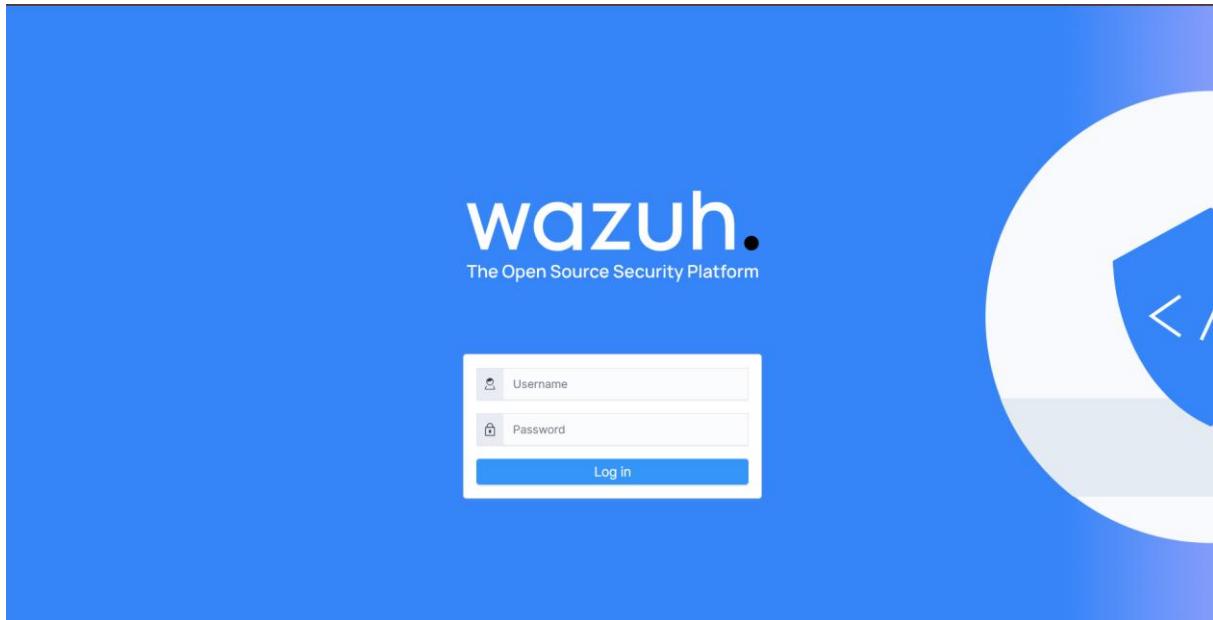


Figure 6 : Page de connexion Wazuh

Page de connexion au tableau de bord Wazuh accessible via HTTPS, demandant les identifiants administrateur.

Figure 7 : Interface du tableau de bord Wazuh

*Interface principale du tableau de bord Wazuh après authentification, présentant les widgets par défaut et les statistiques globales.*

## 4. ENRÔLEMENT DES AGENTS

L'enrôlement des agents constitue une étape critique pour étendre les capacités de monitoring aux endpoints. Cette section détaille le processus d'installation et de configuration des agents Wazuh sur les systèmes Linux et Windows, ainsi que l'intégration de Sysmon pour enrichir la télémétrie Windows.

### 4.1 Agent Linux

Le Dashboard Wazuh fournit un assistant d'enrôlement automatique qui génère les commandes d'installation personnalisées en fonction du système d'exploitation cible.

L'assistant génère automatiquement quatre commandes shell adaptées à notre configuration. Cette approche élimine les erreurs de configuration manuelle et accélère considérablement le déploiement.

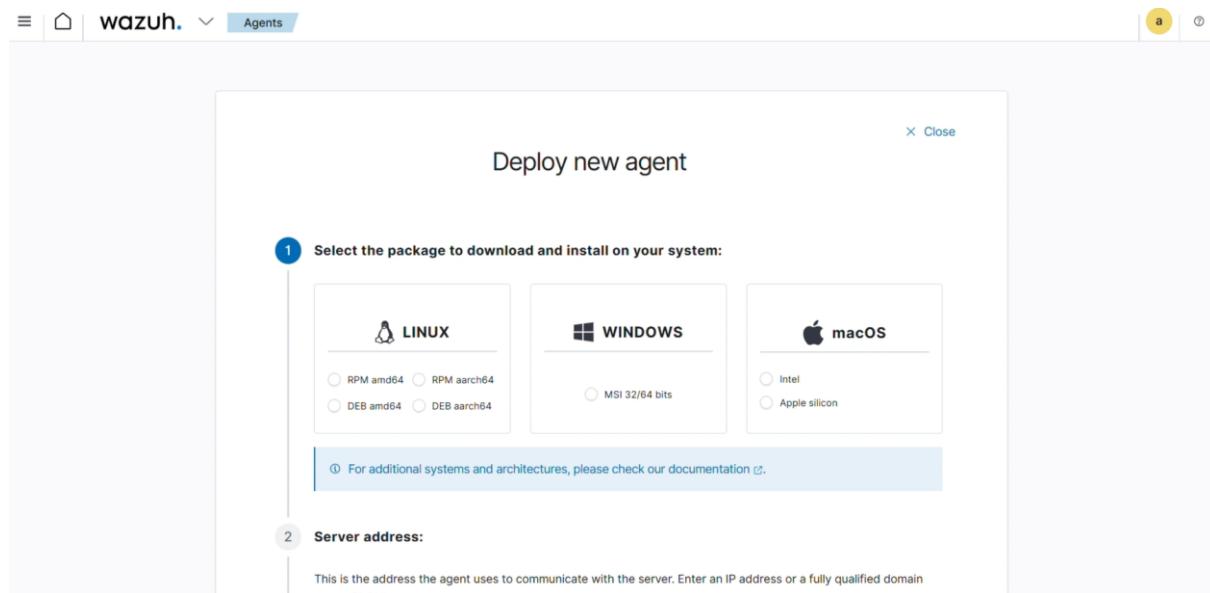


Figure 8 : Interface "Deploy new agent"

*Interface "Deploy new agent" du dashboard Wazuh, permettant de générer les commandes d'installation personnalisées pour l'agent Linux.*

Sur le Linux-Client, nous exécutons les commandes copiées.

```
ssh -i wazuh-key.pem ubuntu@98.91.206.16
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.0-1_amd64.deb
```

```
sudo WAZUH_MANAGER='18.213.113.165' WAZUH_AGENT_NAME='Linux-Client' dpkg -i ./wazuh-agent_4.7.0-1_amd64.deb

sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent
```

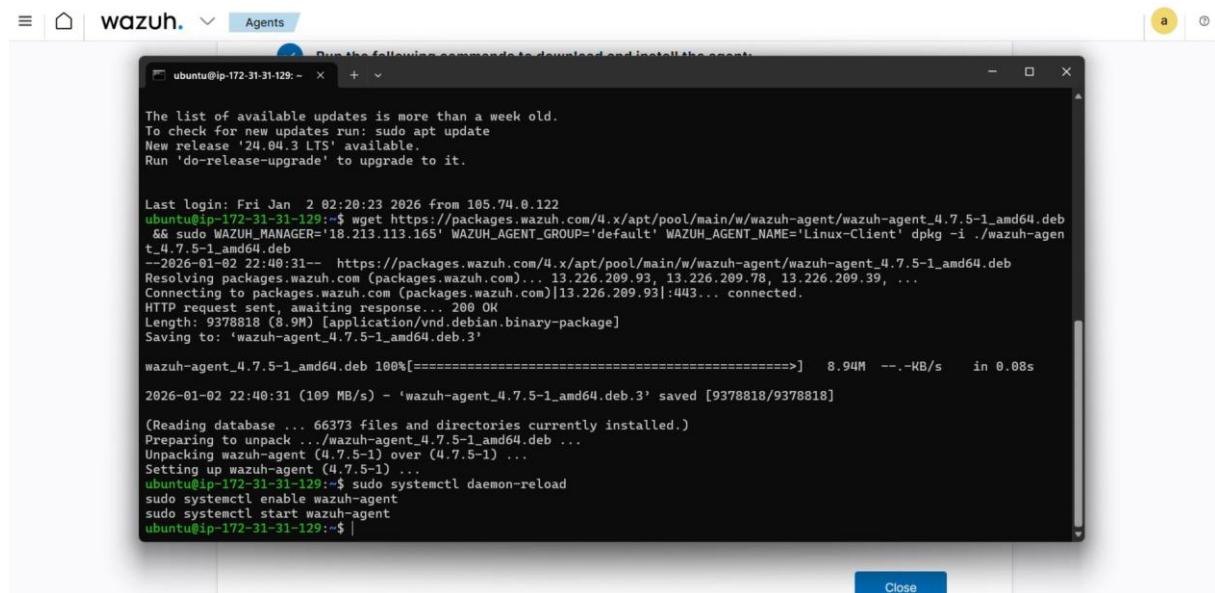
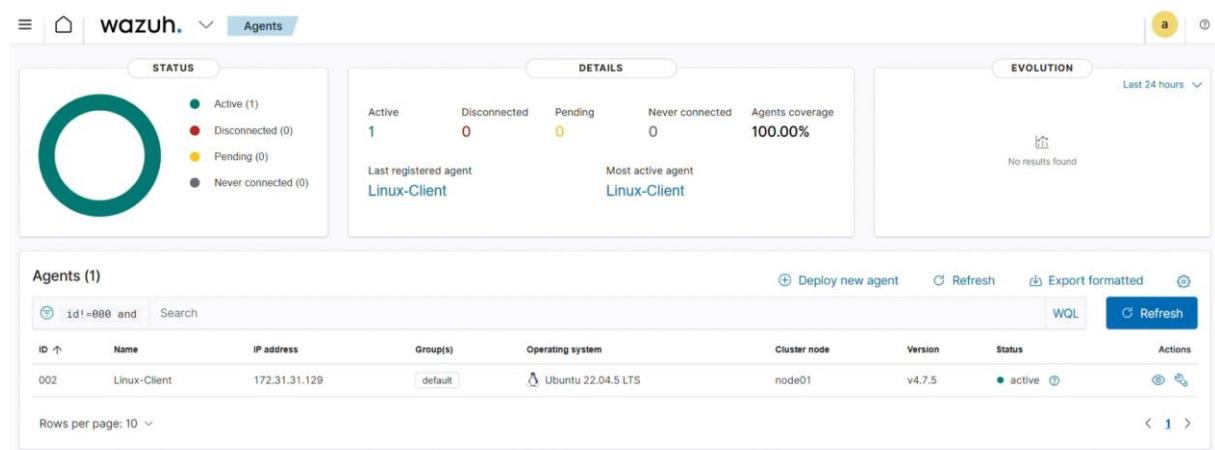


Figure 9 : Installation de l'agent Wazuh sur le Linux-Client

Installation de l'agent Wazuh sur le Linux-Client via SSH, montrant le téléchargement du package .deb et son installation avec les variables d'environnement.



ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
002	Linux-Client	172.31.31.129	default	Ubuntu 22.04.5 LTS	node01	v4.7.5	active	<a href="#">Edit</a> <a href="#">Logs</a>

Figure 10 : Linux-Clien "Active"

*Vue du dashboard Wazuh confirmant l'agent Linux-Client comme "Active" avec ses informations système (OS, version, IP).*

## 4.2 Agent Windows + Sysmon

L'enrôlement de l'agent Windows suit une méthodologie similaire, avec l'ajout de l'intégration Sysmon pour enrichir considérablement la visibilité au niveau système.

### Connexion RDP au Windows-Client

Nous nous connectons à l'instance Windows Server via Remote Desktop Protocol (RDP) ,

Pour les instances Windows sur AWS, le mot de passe Administrator doit être décrypté via la console EC2 en utilisant la clé privée. Cette étape garantit que seul le propriétaire de la clé peut accéder à l'instance.

### Installation de l'agent Wazuh

Depuis le Windows-Client, nous ouvrons PowerShell en mode Administrateur (clic droit → "Exécuter en tant qu'administrateur"). Le Dashboard Wazuh génère les commandes d'installation spécifiques à Windows :

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.0-1.msi -OutFile ${env:tmp}\wazuh-agent.msi
msiexec.exe /i ${env:tmp}\wazuh-agent.msi /q WAZUH_MANAGER='18.213.113.165'
WAZUH_AGENT_NAME='Windows-Client'

NET START WazuhSvc
```

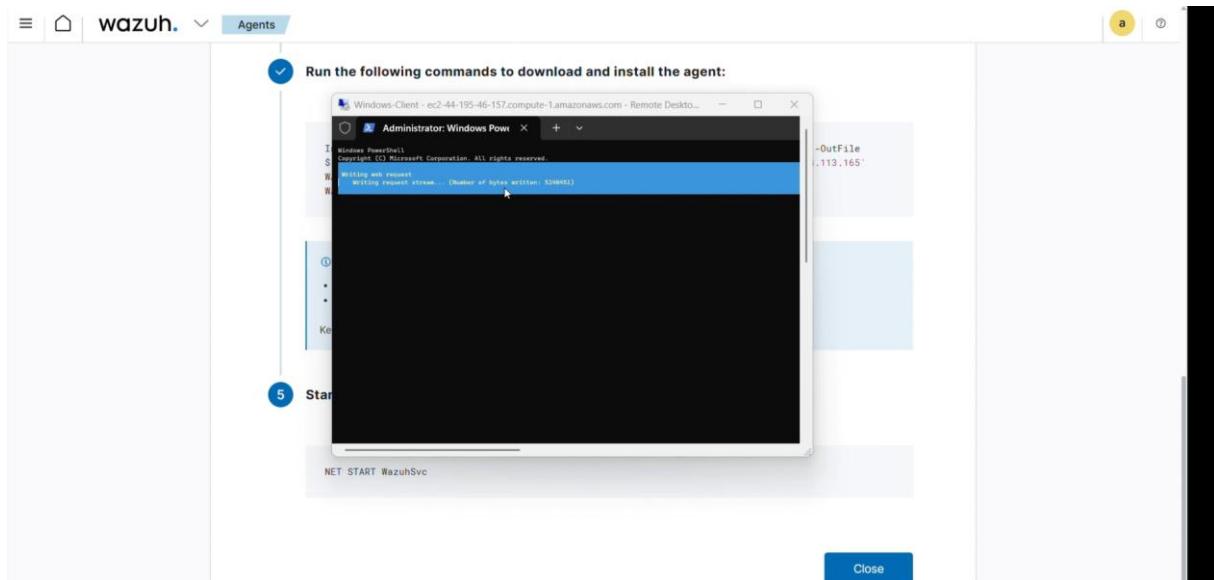


Figure 11 : Installation de l'agent Wazuh sur le Windows-Client

Installation de l'agent Wazuh sur Windows via PowerShell en mode administrateur, utilisant le package MSI et les paramètres de configuration.

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
002	Linux-Client	172.31.31.129	default	Ubuntu 22.04.5 LTS	node01	v4.7.5	active	<a href="#">Details</a>
003	Windows-Client	172.31.3.15	default	Microsoft Windows Server 2025 Datacenter 10.0.26100.7462	node01	v4.7.5	active	<a href="#">Details</a>

Figure 12 : l'agent Windows-Client "Active"

Vue du dashboard Wazuh confirmant l'agent Windows-Client comme "Active" avec ses informations système (OS, version, IP).

## Installation de Sysmon

Sysmon (System Monitor) est un service Windows développé par Microsoft Sysinternals qui enregistre les activités système détaillées dans le journal d'événements. Son intégration avec Wazuh transforme Windows en plateforme EDR robuste.

## Téléchargement et extraction

Nous créons un répertoire dédié et téléchargeons Sysmon :

```
New-Item -Path "C:\Sysmon" -ItemType Directory -Force
cd C:\Sysmon
Invoke-WebRequest -Uri "https://download.sysinternals.com/files/Sysmon.zip" -OutFile "Sysmon.zip"
Expand-Archive -Path "Sysmon.zip" -DestinationPath "C:\Sysmon"
Invoke-WebRequest -Uri "https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml" -OutFile "sysmonconfig.xml"
.\Sysmon64.exe -accepteula -i sysmonconfig.xml
```

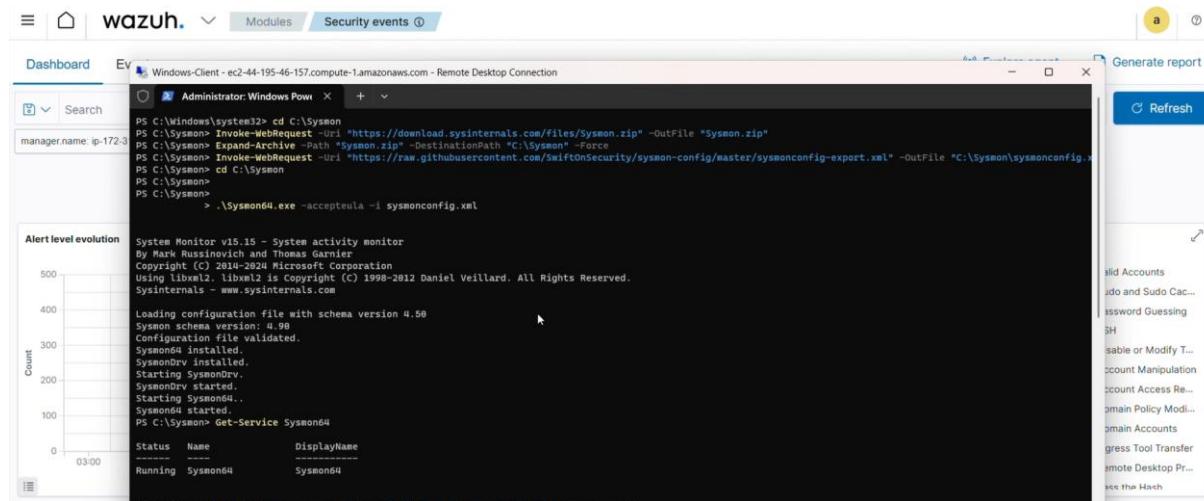


Figure 13 : Installation et configuration de Sysmon

*Installation et configuration de Sysmon sur Windows-Client via PowerShell, incluant le téléchargement, l'extraction et l'application d'un fichier de configuration.*

L'intégration Sysmon+Wazuh fournit ainsi une visibilité au niveau kernel Windows, permettant la détection de techniques avancées qui échappent aux antivirus traditionnels basés sur signatures.

## PARTIE IV : TESTS ET ANALYSES

### 1. SCÉNARIOS DE SÉCURITÉ TESTÉS

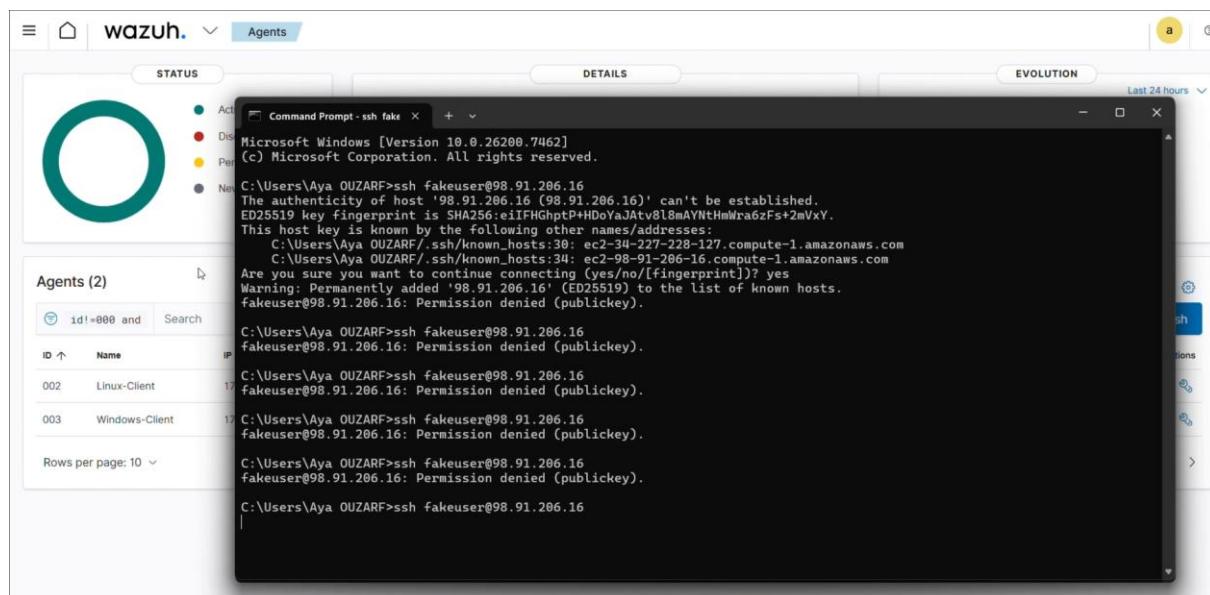
Cette section présente l'ensemble des scénarios d'attaque simulés pour valider les capacités de détection de la plateforme Wazuh. Chaque scénario est documenté avec les commandes exécutées, les événements générés, et l'analyse des alertes produites.

#### 1.1 SSH Bruteforce (Linux)

L'attaque par force brute SSH constitue l'une des menaces les plus courantes contre les serveurs Linux exposés sur Internet. Les attaquants utilisent des dictionnaires de mots de passe ou des attaques par combinaison pour tenter de deviner les credentials SSH et obtenir un accès initial au système.

Depuis un poste externe, nous simulons une série de tentatives de connexion SSH avec des identifiants invalides vers le Linux-Client :

```
ssh fakeuser@98.91.206.16
```



The screenshot shows the Wazuh interface with the 'Agents' tab selected. A terminal window is open, showing multiple failed SSH connection attempts from an external host (98.91.206.16) to a Linux-Client (IP 172.17.0.2). The logs indicate that the host key fingerprint could not be established and that permission was denied for each attempt. The terminal session is titled 'Command Prompt - ssh fake'.

```

Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aya OUZARF>ssh fakeuser@98.91.206.16
The authenticity of host '98.91.206.16 (98.91.206.16)' can't be established.
ED25519 key fingerprint is SHA256:eIIfHGhpt+HDoVaAtvb18mAVNtHmWradzFs+2mVXY.
This host key is known by the following other names/addresses:
  C:\Users\Aya OUZARF/.ssh/known_hosts:30: ec2-34-227-228-127.compute-1.amazonaws.com
  C:\Users\Aya OUZARF/.ssh/known_hosts:34: ec2-98-91-206-16.compute-1.amazonaws.com

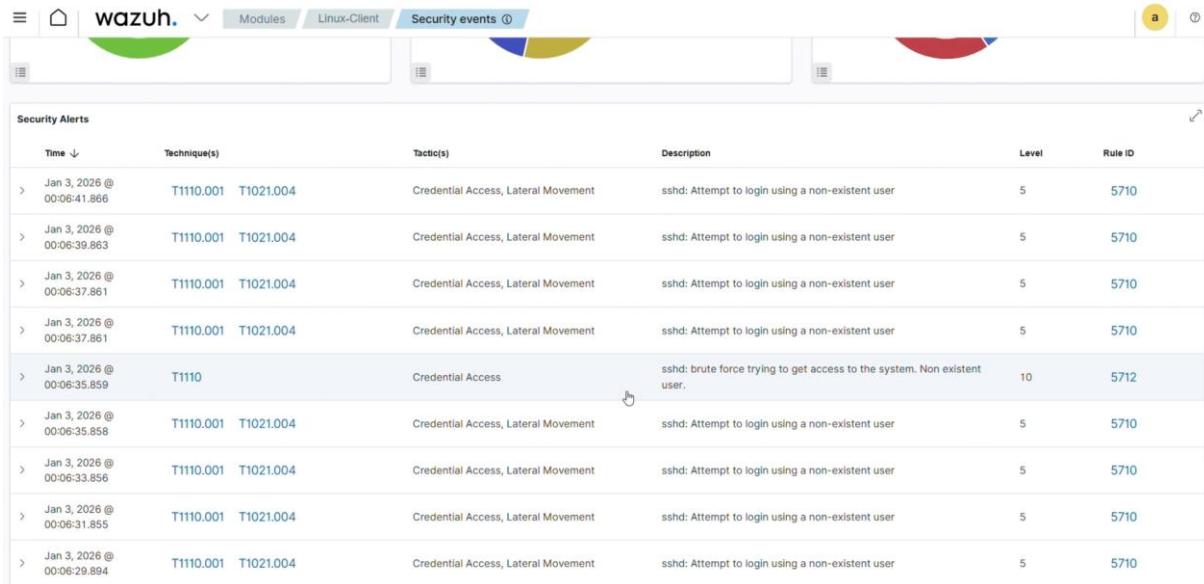
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.91.206.16' (ED25519) to the list of known hosts.
fakeuser@98.91.206.16: Permission denied (publickey).

C:\Users\Aya OUZARF>ssh fakeuser@98.91.206.16
fakeuser@98.91.206.16: Permission denied (publickey).

```

Figure 14 : Tentatives SSH échouées

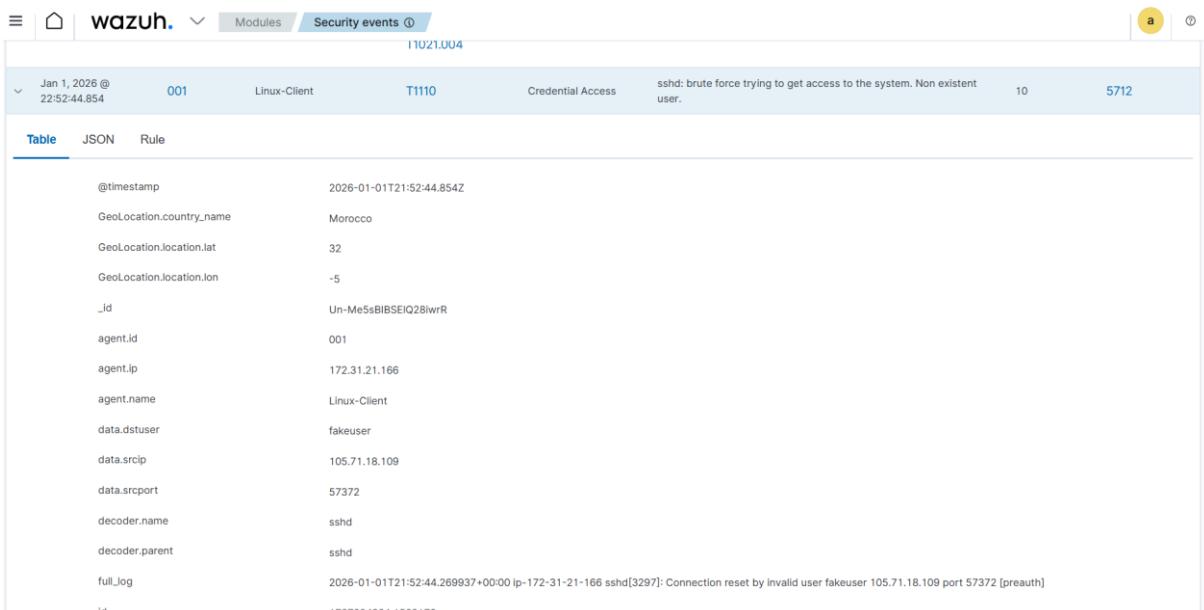
Tentatives SSH échouées depuis un terminal externe vers le Linux-Client, simulant une attaque par force brute avec des identifiants invalides.



Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 3, 2026 @ 00:06:41.866	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:39.863	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:37.861	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:37.861	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:35.859	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712
Jan 3, 2026 @ 00:06:35.858	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:33.856	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:31.855	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Jan 3, 2026 @ 00:06:29.894	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710

Figure 15 : Les alerte SSH brute force

Tableau de bord Wazuh affichant les alertes générées par les tentatives SSH échouées, visibles dans la vue "Security events"



Field	Value
@timestamp	2026-01-01T21:52:44.854Z
GeoLocation.country_name	Morocco
GeoLocation.location.lat	32
GeoLocation.location.lon	-5
_id	Un-Me5sBIBSEIQ28lwR
agent.id	001
agent.ip	172.31.21.166
agent.name	Linux-Client
data.dstuser	fakeuser
data.srcip	105.71.18.109
data.srport	57372
decoder.name	sshd
decoder.parent	sshd
full_log	2026-01-01T21:52:44.269937+00:00 ip-172-31-21-166 sshd[329]: Connection reset by invalid user fakeuser 105.71.18.109 port 57372 [preauth]
id	1767304364.1563170

Figure 16 : Détails d'une alerte SSH brute force

Détails d'une alerte SSH brute force dans Wazuh, montrant la règle déclenchée (5710), l'IP source, le nom d'utilisateur tenté et les métadonnées de l'événement.

L'alerte de sévérité **Medium (5)** signale une tentative de bruteforce, justifiant des actions automatiques : blocage temporaire de l'IP, renforcement du monitoring, notification au SOC et suivi de l'IP. La règle **5710** de Wazuh montre l'efficacité immédiate de la solution sans configuration préalable.

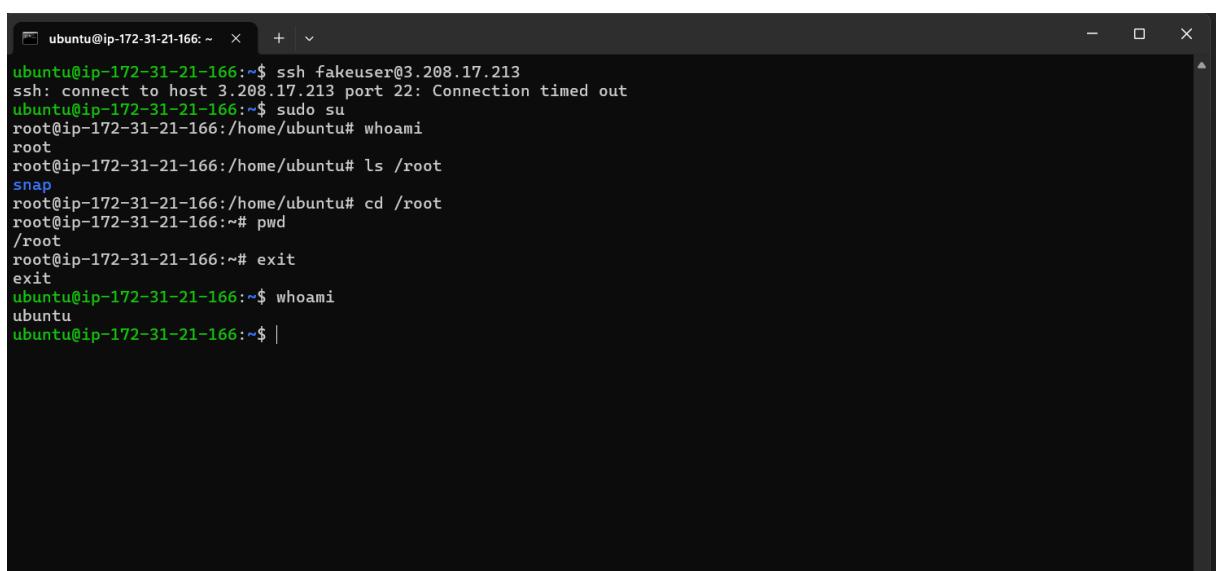
## 1.2 Élévation de privilèges et FIM (Linux)

Cette section couvre deux aspects critiques de la sécurité Linux : la surveillance des élévations de privilèges via sudo et la détection des modifications de fichiers sensibles via File Integrity Monitoring (FIM).

### Scénario Sudo - Élévation de privilèges

La commande sudo permet aux utilisateurs autorisés d'exécuter des commandes avec les privilèges root.

Sur le Linux-Client, nous exécutons plusieurs commandes administratives via sudo :



```
ubuntu@ip-172-31-21-166:~$ ssh fakeuser@3.208.17.213
ssh: connect to host 3.208.17.213 port 22: Connection timed out
ubuntu@ip-172-31-21-166:~$ sudo su
root@ip-172-31-21-166:/home/ubuntu# whoami
root
root@ip-172-31-21-166:/home/ubuntu# ls /root
snap
root@ip-172-31-21-166:/home/ubuntu# cd /root
root@ip-172-31-21-166:~/# pwd
/root
root@ip-172-31-21-166:~/# exit
exit
ubuntu@ip-172-31-21-166:~$ whoami
ubuntu
ubuntu@ip-172-31-21-166:~$ |
```

Figure 17 : Exécution de la commande sudo su

Exécution de la commande sudo su sur le Linux-Client, démontrant une élévation de privilèges vers l'utilisateur root.

Time	User	Host	Source	Action	Details	Severity	Rule
23:00:23.197	UUI	Linux-Client	T1U/8	Escalation, Initial Access	PAM: Login session opened.	3	5402
Jan 1, 2026 @ 23:00:23.156	001	Linux-Client	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402

Table JSON Rule

```

@timestamp      2026-01-01T22:00:23.156Z
_id             XH-Te5sBIBSElQ28nAqA
agent.id        001
agent.ip        172.31.21.166
agent.name      Linux-Client
data.command    /usr/bin/su
data.dstuser    root
data.pwd        /home/ubuntu
data.srcuser    ubuntu
data.tty         pts/0
decoder.ftcomment First time user executed the sudo command
decoder.name    sudo
decoder.parent   sudo
full_log        2026-01-01T22:00:21.393231+00:00 ip-172-31-21-166 sudo: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/su

```

Figure 18 : Alerta sudo

Alerte Wazuh générée par l'utilisation de sudo (règle 5402), enregistrant l'élévation de priviléges avec l'utilisateur et la commande exécutée.

L'**agent Wazuh** enregistre les actions sudo, et le **Manager applique la règle 5402** (niveau Low 3) pour les succès vers root, car il s'agit d'une action légitime mais à tracer. Les métadonnées collectées (utilisateur source et cible, commande, terminal, répertoire) permettent un **audit précis** et, en cas de comportements suspects, des alertes plus élevées peuvent être déclenchées via corrélation.

### Scénario FIM - Modification fichier sensible

Le FIM surveille les modifications de fichiers critiques pour détecter des compromissions, backdoors, ou modifications non autorisées. Le fichier /etc/passwd est particulièrement sensible car il contient la liste des comptes utilisateurs du système.

Nous modifions le fichier /etc/passwd en ajoutant un commentaire :

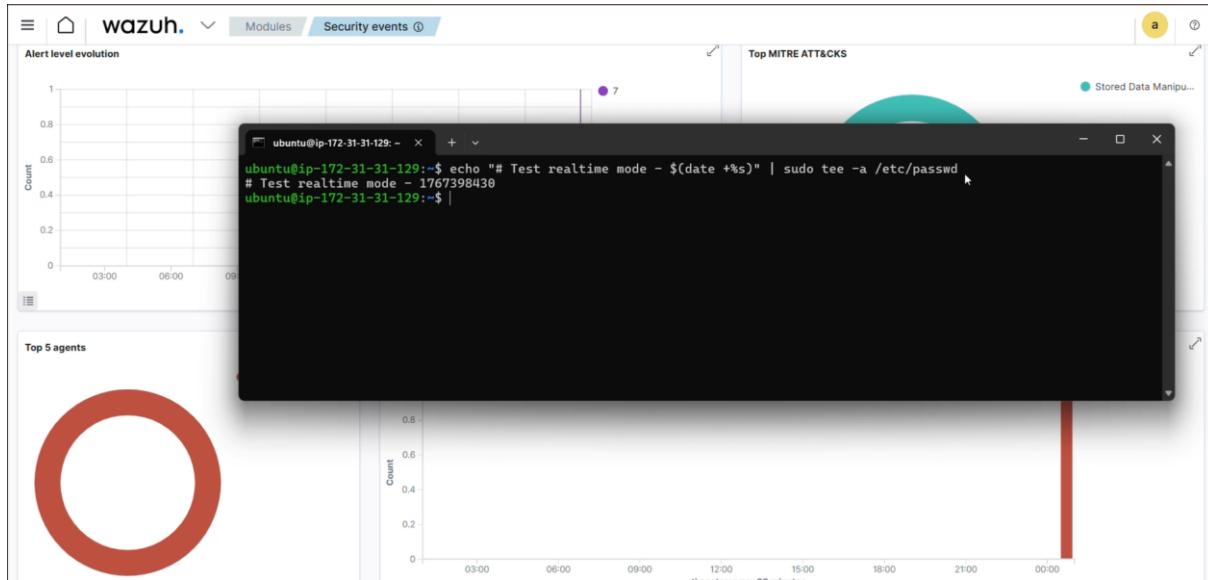


Figure 19 : Modification /etc/passwd

*Modification du fichier sensible /etc/passwd via la commande echo, illustrant un scénario de manipulation de fichier système critique.*

Cette modification est détectée lors du prochain scan FIM (par défaut toutes les 6 heures, mais configurable).

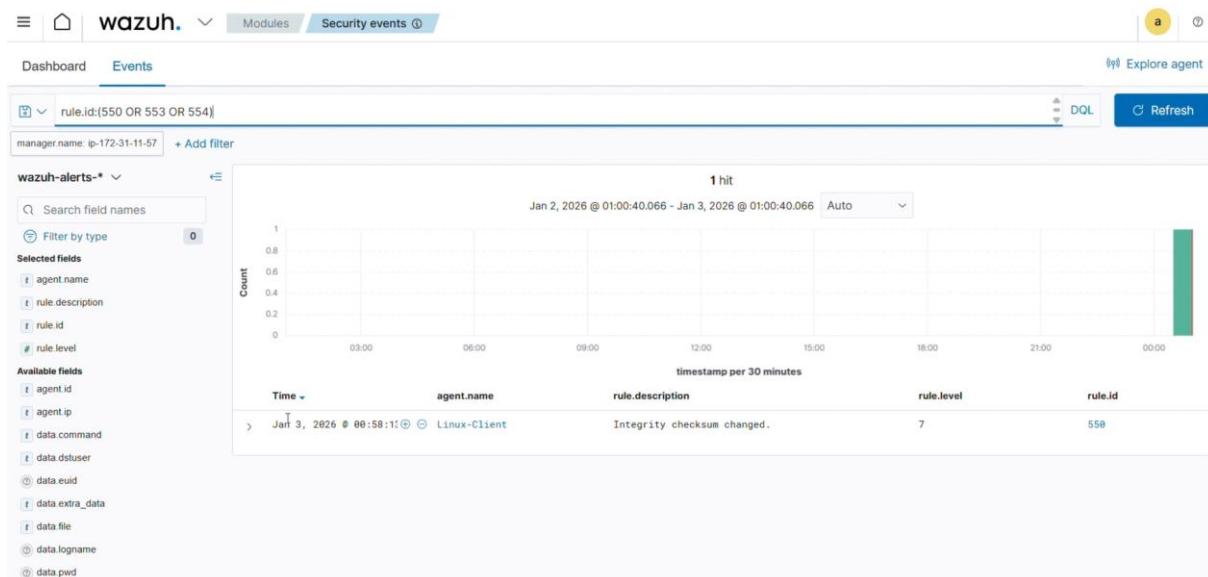
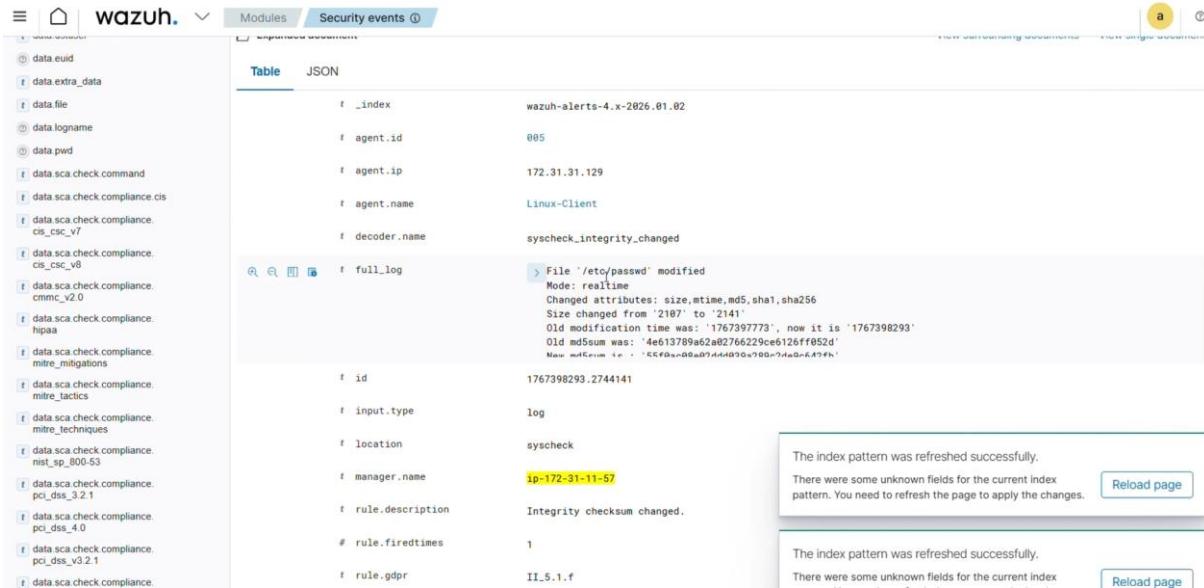


Figure 20 : Alerte File Integrity Monitoring (FIM)

*Alerte File Integrity Monitoring (FIM) dans Wazuh détectant la modification de /etc/passwd avec les hash avant/après et le niveau de严重性 High.*



The screenshot shows the Wazuh Security events interface. On the left, there's a sidebar with various log categories like 'data\_euid', 'data\_extra\_data', etc. The main area is titled 'Security events' and shows a table of log entries. One entry is expanded to show details about a modification to the '/etc/passwd' file. The expanded log entry includes fields such as '\_index', 'agent.id', 'agent.ip', 'agent.name', 'decoder.name', 'full\_log' (which contains a detailed file change message), 'id', 'input.type', 'location', 'manager.name', 'rule.description', 'rule.firedtimes', and 'rule.gdpr'. To the right of the table, there are two boxes: one for the expanded log entry and another for the 'full\_log' field. Both boxes contain messages about index refresh and unknown fields, with a 'Reload page' button.

Figure 21 : Détails complets de l'alerte FIM

Détails complets de l'alerte FIM, incluant les informations de fichier, les empreintes cryptographiques et le contexte de modification.

Le Manager Wazuh déclenche une **alerte FIM (Rule 550, niveau High 7)** pour toute modification du fichier critique /etc/passwd. L'alerte fournit les **hashs avant/après**, permissions et propriétaire, et signale un risque élevé de **backdoor, persistance ou compromission**. En production, elle nécessite une **investigation immédiate** : vérifier la légitimité du changement, consulter les logs, examiner les processus et contrôler l'intégrité d'autres fichiers système.

### Corrélation Sudo + FIM

La combinaison des deux événements (sudo suivi de FIM sur /etc/passwd) dans une fenêtre temporelle courte peut déclencher une règle de corrélation de niveau Critical, indiquant une potentielle compromission en cours.

### 1.3 Échecs de connexion RDP (Windows)

Remote Desktop Protocol (RDP) est le principal vecteur d'accès distant pour les systèmes Windows Server. Les attaquants ciblent massivement le port 3389/TCP avec des attaques bruteforce automatisées pour compromettre les serveurs exposés.

Nous répétons l'opération 5 fois :

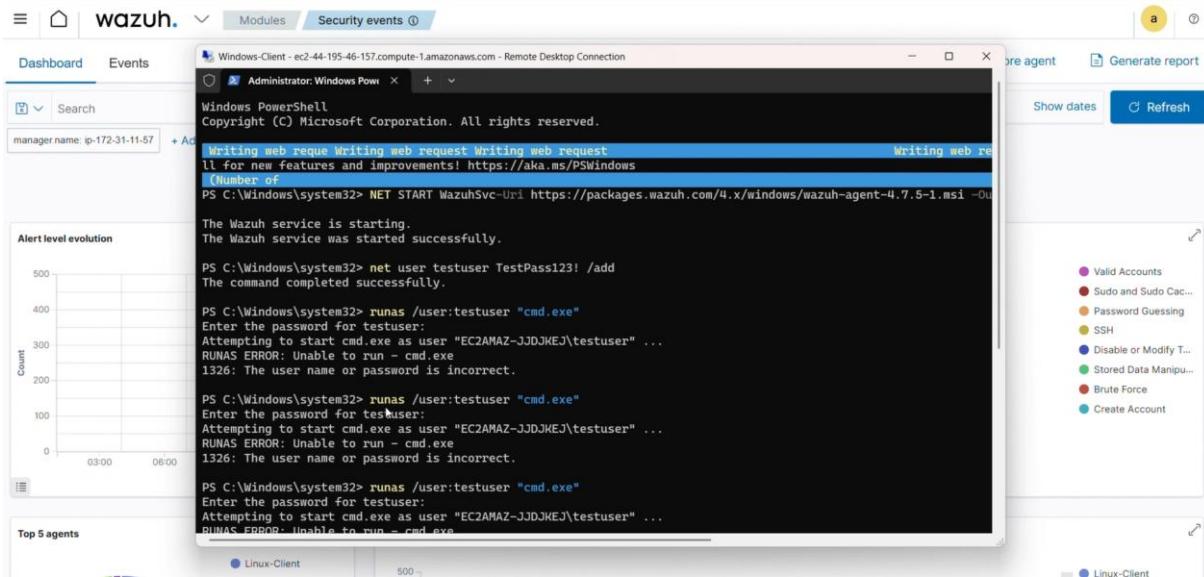


Figure 22 : Tentatives de connexion RDP échouées

Tentatives de connexion RDP échouées vers le Windows-Client, affichant le message d'erreur d'authentification dans l'interface Remote Desktop.

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 3, 2026 @ 01:06:45.266	003	Windows-Client	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
> Jan 3, 2026 @ 01:06:32.315	003	Windows-Client	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
> Jan 3, 2026 @ 01:06:27.461	003	Windows-Client	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
> Jan 3, 2026 @ 01:06:24.384	003	Windows-Client	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
> Jan 3, 2026 @ 01:05:42.927	003	Windows-Client	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
> Jan 3, 2026 @ 01:03:10.303	003	Windows-Client	T1484	Defense Evasion, Privilege Escalation	Users group changed.	5	60170
> Jan 3, 2026 @ 01:03:10.300	003	Windows-Client	T1098	Persistence	User account changed.	8	60110
> Jan 3, 2026 @ 01:03:10.298	003	Windows-Client	T1098	Persistence	User account enabled or created.	8	60109

Figure 23 : Alert de connexion RDP échouées

Journal des événements Windows (Event Viewer) montrant les événements de sécurité 4625 (échecs de connexion) avec les détails d'authentification.



The screenshot shows a list of Windows event logs under the 'Security events' tab. One event is highlighted, showing details of a failed logon attempt:

- data.win.eventdata.targetUserName:** testuser
- data.win.eventdata.targetUserSid:** S-1-0-0
- data.win.eventdata.workstationName:** EC2AMAZ-JJDKEJ
- data.win.system.channel:** Security
- data.win.system.computer:** EC2AMAZ-JJDKEJ
- data.win.system.eventID:** 4625
- data.win.system.eventRecordID:** 84582
- data.win.system.keywords:** 0x8010000000000000
- data.win.system.level:** 0
- data.win.system.message:** "An account failed to log on.

Subject:  
 Security ID: S-1-5-21-315438573-3231945463-2044557683-500  
 Account Name: Administrator  
 Account Domain: EC2AMAZ-JJDKEJ

Figure 24 : Détail d'alerte connexion RDP échouées

Alerte Wazuh correspondant aux échecs RDP, affichant l'Event ID 4625, le type de connexion (Logon Type 10) et les informations de compte.

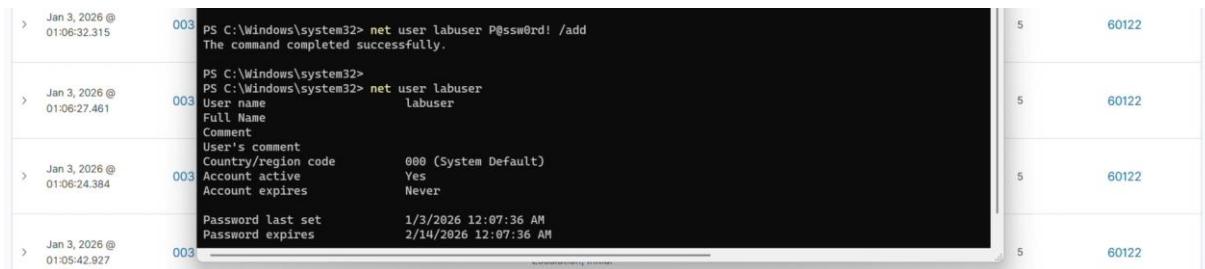
L'agent Wazuh collecte les **Event ID 4625** et le **Manager déclenche la règle 60122 (Medium 5)** après plusieurs échecs depuis la même IP, signalant une **tentative de bruteforce (MITRE T1110.001)**. L'alerte inclut **IP source, compte cible, type de connexion, raison de l'échec et nombre de tentatives**, et le **Dashboard** permet de visualiser ces données via timeline, géolocalisation et classements des comptes et IPs les plus ciblés.

## 1.4 Crédit d'utilisateur Windows

La création d'un compte utilisateur avec priviléges administrateurs est une technique classique de persistence utilisée par les attaquants après compromission initiale. Cette action génère des Event IDs Windows spécifiques qui doivent être surveillés avec un niveau de criticité élevé.

Sur le Windows-Client, nous ouvrons PowerShell en mode Administrateur et exécutons les commandes suivantes :

```
net user labuser P@ssw0rd! /add
net localgroup administrators labuser /add
```



```

> Jan 3, 2026 @ 01:06:32.315      003 PS C:\Windows\system32> net user labuser P@ssw0rd! /add
The command completed successfully.

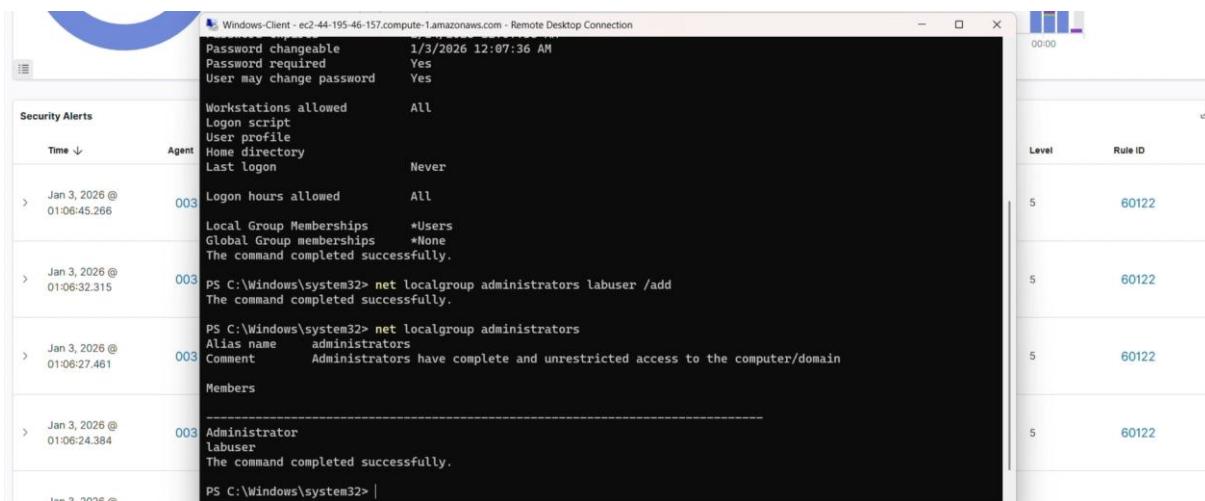
> Jan 3, 2026 @ 01:06:27.461      003 PS C:\Windows\system32> net user labuser
User name          labuser
Full Name
Comment
User's comment
Country/Region code    000 (System Default)
Account active        Yes
Account expires       Never

Password last set    1/3/2026 12:07:36 AM
Password expires      2/14/2026 12:07:36 AM

> Jan 3, 2026 @ 01:06:24.384      003
> Jan 3, 2026 @ 01:05:42.927      003

```

Figure 25 : Création d'un utilisateur local "labuser"



```

Windows-Client - ec2-44-195-46-157.compute-1.amazonaws.com - Remote Desktop Connection
Password changeable      1/3/2026 12:07:36 AM
Password required        Yes
User may change password Yes

Security Alerts
Time ↓
Agent
Workstations allowed     All
Logon script
User profile
Home directory
Last logon               Never

> Jan 3, 2026 @ 01:06:45.266      003 Logon hours allowed     All
Local Group Memberships   *Users
Global Group memberships *None
The command completed successfully.

> Jan 3, 2026 @ 01:06:32.315      003 PS C:\Windows\system32> net localgroup administrators labuser /add
The command completed successfully.

> Jan 3, 2026 @ 01:06:27.461      003 PS C:\Windows\system32> net localgroup administrators
Alias name               administrators
Comment                 Administrators have complete and unrestricted access to the computer/domain
Members

> Jan 3, 2026 @ 01:06:24.384      003 Administrator
labuser
The command completed successfully.

PS C:\Windows\system32>

```

Figure 26 : Ajout l'utilisateur au groupe Administrateurs

*Création d'un utilisateur local "labuser" et ajout au groupe Administrateurs via PowerShell avec les commandes net user et net localgroup.*

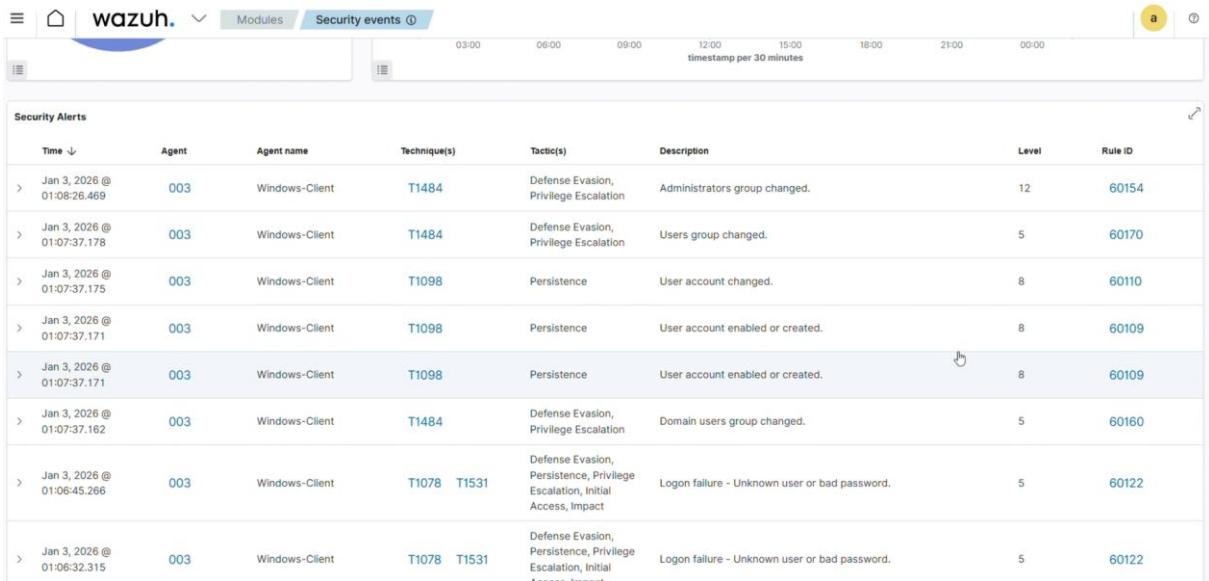


Figure 27 : Alert l'ajout d'un utilisateur

*Journal des événements Windows confirmant la création du compte utilisateur (Event ID 4720) avec le nom du compte et l'utilisateur ayant effectué l'action. et Alerte Wazuh détectant l'ajout d'un utilisateur à un groupe privilégié (règle 60132) avec un niveau de严重性 Critical*

Le Manager Wazuh corrèle plusieurs événements et déclenche des alertes en cascade :

- **Création de compte (Rule 60106, High 8)** – MITRE T1136.001, signale un nouvel utilisateur local.
- **Ajout au groupe Administrators (Rule 60132, Critical 9)** – MITRE T1078.003, critique car le compte backdoor permet **persistence, privilèges totaux, discréption et mouvement latéral** dans le réseau.

## 1.5 Événements Sysmon (EDR Windows)

Cette section démontre les capacités EDR avancées offertes par l'intégration Sysmon avec Wazuh. Nous présentons plusieurs catégories d'événements capturés et leur pertinence pour la détection de menaces.

Avant de générer des événements de test, nous validons que le service Sysmon est opérationnel :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> Get-Service Sysmon64

Status     Name           DisplayName
-----     --           -----
Running    Sysmon64      Sysmon64

PS C:\Windows\system32>
```

Figure 28 : Service Sysmon64 actif

### Service Sysmon64 actif et en cours d'exécution

Nous vérifions également que Wazuh collecte bien les événements Sysmon en consultant le Dashboard, section "Security events", avec le filtre :

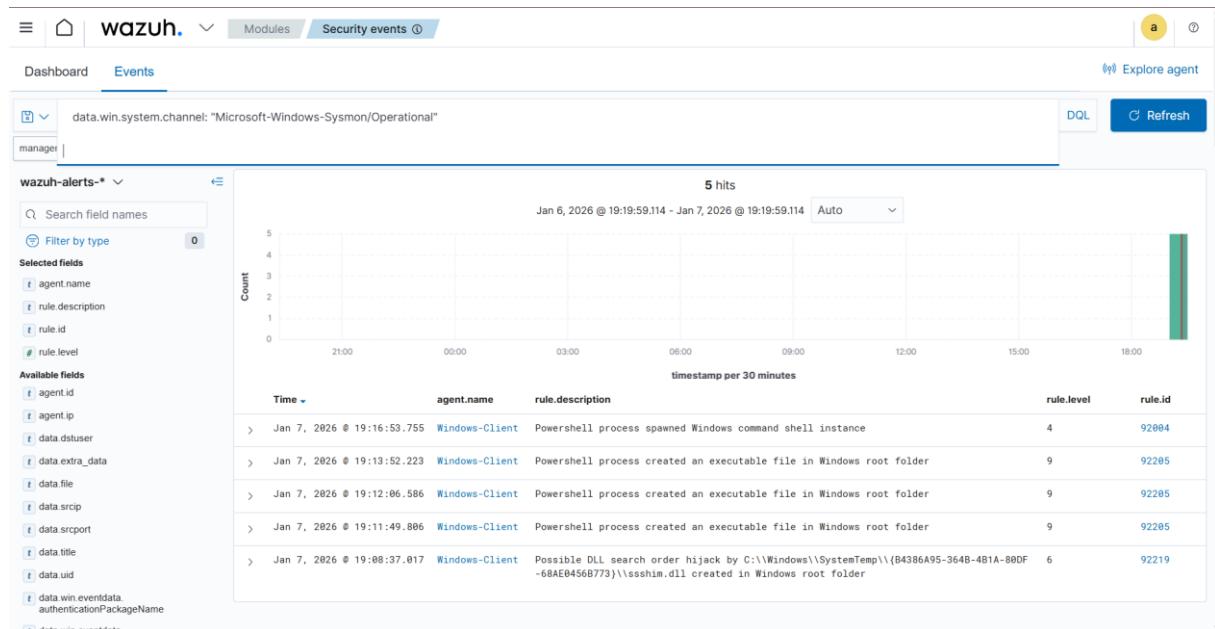


Figure 29 : Collection des événements Sysmon

"Wazuh collecte bien les événements Sysmon"

### Test 1 : Détection de création de processus (Event ID 1)

L'Event ID 1 de Sysmon capture chaque création de processus avec des métadonnées détaillées : ligne de commande complète, processus parent, hashes du fichier exécutables, utilisateur, répertoire de travail. Cette visibilité est essentielle pour détecter l'exécution de malware ou de commandes suspectes.

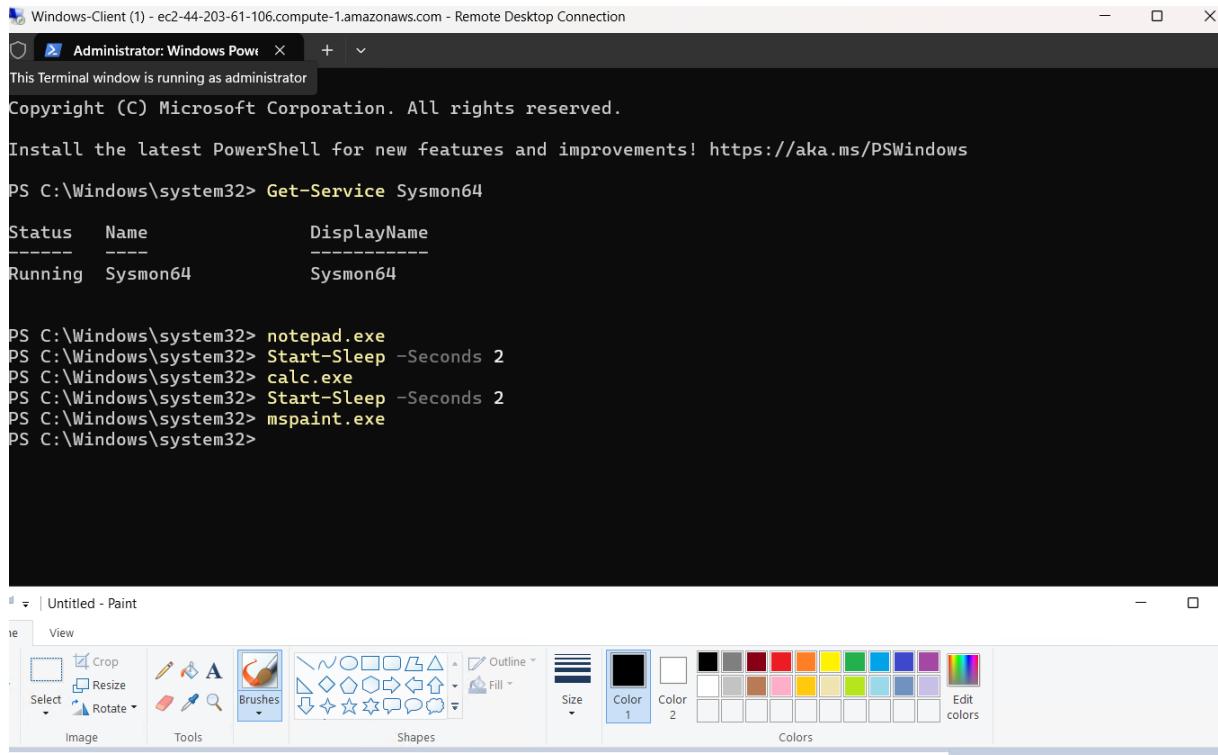


Figure 30 : L'Event ID 1

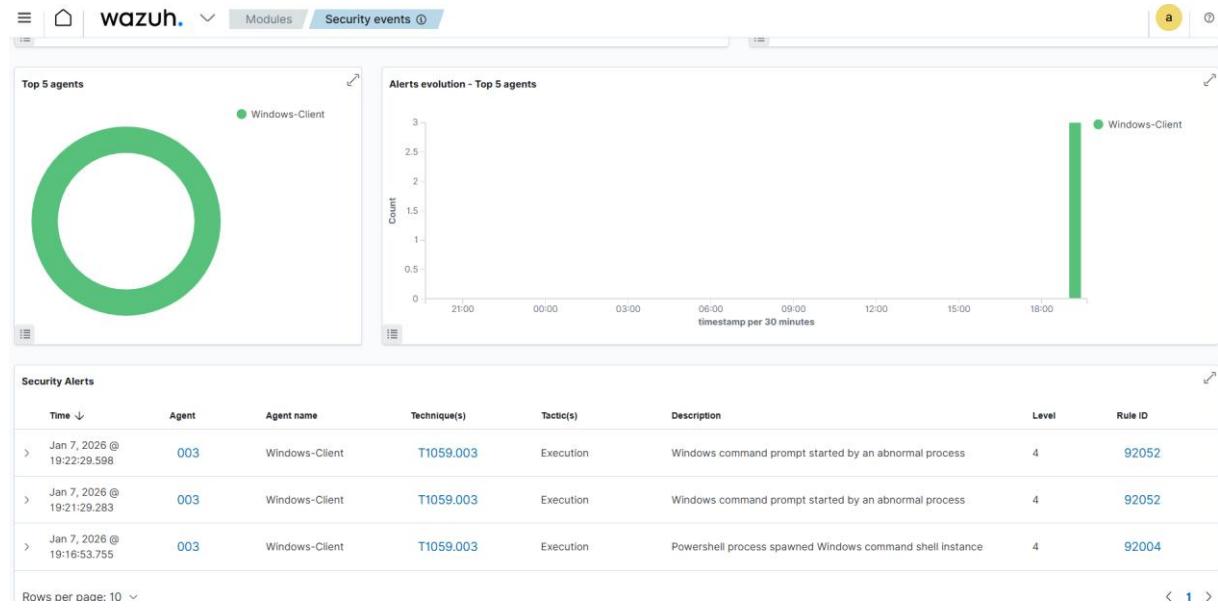


Figure 31 : Alert d'Event ID 1

"Sysmon Event ID 1 - Détection des processus notepad, calc, mspaint"

### Test 2 : Détection de création de fichiers (Event ID 11)

L'Event ID 11 capture chaque création de fichier, permettant de détecter les ransomwares (création massive de fichiers .encrypted), les droppers (dépose de payload), ou les modifications suspectes.

Nous créons plusieurs fichiers de test

Mode	LastWriteTime	Length	Name
---	-----	-----	-----
-a---	1/7/2026 6:42 PM	42	suspicious_file.exe.txt
-a---	1/3/2026 12:43 AM	0	sysmon_test_004323.txt
-a---	1/7/2026 6:42 PM	26	test1.txt
-a---	1/7/2026 6:42 PM	26	test2.txt
-a---	1/7/2026 6:17 PM	9	testfile.txt

Figure 32 : L'Event ID 11

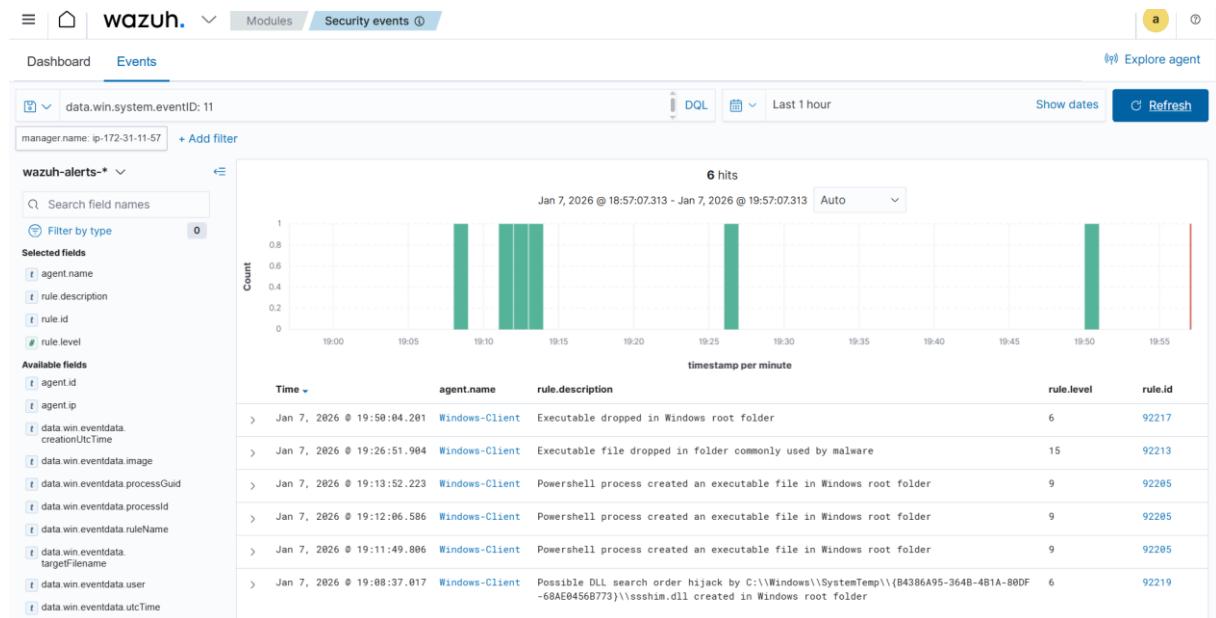


Figure 33 : Alert d'Event ID 11

"Sysmon Event ID 11 - Fichiers créés dans C:\temp détectés"

### Test 3 : Détection de connexions réseau (Event ID 3)

L'Event ID 3 enregistre chaque connexion réseau sortante, permettant de détecter les communications C2 (Command and Control), l'exfiltration de données, ou les connexions vers des domaines malveillants.

Depuis PowerShell, nous générerons des connexions réseau vers plusieurs sites

```
Windows-Client (1) - ec2-44-203-61-106.compute-1.amazonaws.com - Remote Desktop Connection

PS C:\Sysmon> Test-NetConnection 8.8.8.8 -Port 53

ComputerName      : 8.8.8.8
RemoteAddress     : 8.8.8.8
RemotePort        : 53
InterfaceAlias    : Ethernet
SourceAddress     : 172.31.3.15
TcpTestSucceeded  : True

PS C:\Sysmon> Test-NetConnection google.com -Port 443

ComputerName      : google.com
RemoteAddress     : 142.251.163.113
RemotePort        : 443
InterfaceAlias    : Ethernet
SourceAddress     : 172.31.3.15
TcpTestSucceeded  : True

PS C:\Sysmon> ping microsoft.com

Pinging microsoft.com [13.107.246.41] with 32 bytes of data:
Request timed out.
```

Figure 34 : L'Event ID 3

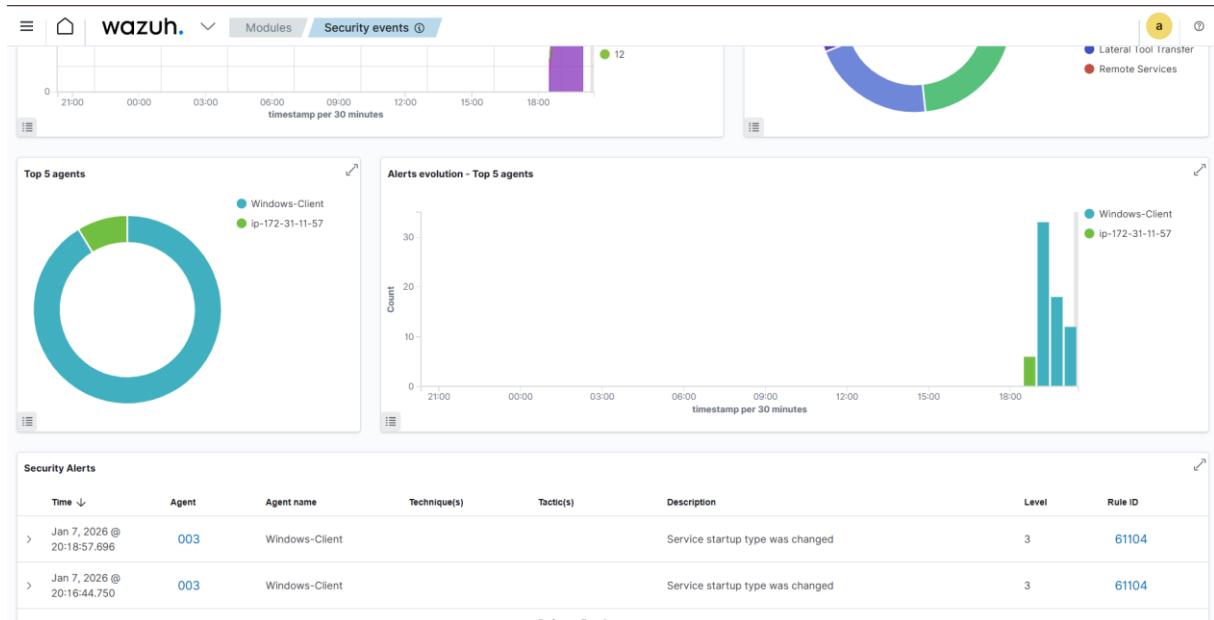


Figure 35 : Alert d'Event ID 3

"Sysmon Event ID 3 - Connexions réseau détectées vers google.com et autres"

### Test 4 : Simulation d'activité PowerShell suspecte

Déetecter l'utilisation de PowerShell avec des techniques d'obfuscation ou de téléchargement de code, couramment utilisées par les attaquants.

```
PS C:\Sysmon> New-Item -Path "C:\temp\suspicious" -ItemType Directory -Force

Directory: C:\temp

Mode                LastWriteTime       Length Name
----                -----          ----
d----- 1/7/2026 7:08 PM           0    suspicious

PS C:\Sysmon> Copy-Item "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" "C:\temp\suspicious\ps.exe"
PS C:\Sysmon> C:\temp\suspicious\ps.exe -Command "Write-Host 'Test EDR Detection'"
Test EDR Detection
PS C:\Sysmon>
```

Figure 36 : Activité PowerShell suspecte

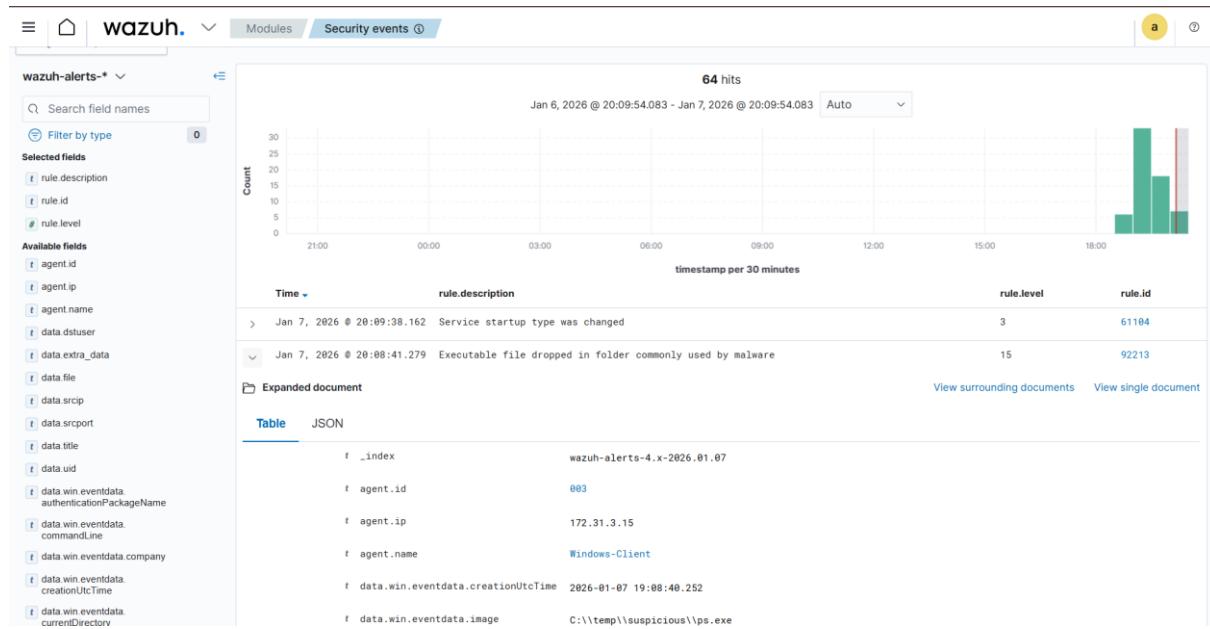


Figure 37 : Alert d'activité PowerShell suspecte

"Détection d'activité PowerShell suspecte - Level 9 critique"

### Synthèse des tests Sysmon

Les tests réalisés démontrent la puissance de l'intégration Sysmon + Wazuh pour transformer Windows en plateforme EDR complète. La visibilité au niveau kernel permet de détecter des techniques avancées que les antivirus traditionnels manqueraient :

Catégorie	Event IDs testés	Utilité principale
Processus	1, 5	Détection de malware, analyse chaîne d'exécution
Fichiers	11	Détection ransomware, droppers
Réseau	3	Détection C2, exfiltration
Commandes	1 (PowerShell)	Détection scripts malveillants, living off the land

L'approche EDR comportementale complète l'approche SIEM basée sur logs, offrant une défense en profondeur.

## 2. ANALYSE COMPARATIVE SIEM VS EDR

### 2.1 Différences fondamentales

Le SIEM et l'EDR sont deux approches complémentaires qui offrent des perspectives différentes sur la sécurité. Le SIEM fournit une vue d'ensemble de toute l'infrastructure en collectant et corrélant des événements de multiples sources. L'EDR offre une visibilité détaillée et continue au niveau de chaque endpoint individuel.

Tableau comparatif SIEM vs EDR

Critère	SIEM (Wazuh)	EDR (Sysmon + Wazuh)
Scope	Infrastructure globale	Endpoint spécifique
Sources	Multi-sources (OS, apps, réseau, sécurité)	Endpoint uniquement
Approche	Corrélation de logs	Détection comportementale
Granularité	Événements système	Processus, threads, injections
Force principale	Vue d'ensemble, conformité	Visibilité détaillée, forensics
Use case typique	Déetecter bruteforce multi-systèmes	Analyser chaîne d'exécution malware
Investigation	Corrélation événements	Timeline détaillée endpoint
False positives	Plus élevés sans tuning	Moins fréquents (comportemental)
Couverture	Large (toute l'infra)	Profonde (un endpoint)

## 2.2 Complémentarité démontrée

Dans notre lab, la complémentarité SIEM/EDR est clairement visible. Pour le scénario SSH bruteforce, le SIEM Wazuh a détecté le pattern global (10 tentatives en 5 minutes) en corrélant les logs d'authentification. Cependant, il ne montre pas ce qui se passe au niveau système si l'attaque réussit. À l'inverse, Sysmon sur Windows a capturé précisément la chaîne d'exécution lors de la création de l'utilisateur malveillant : quel processus a lancé net.exe, avec quels arguments, depuis quel compte. Cette visibilité détaillée est impossible avec le SIEM seul.

## 2.3 Cas d'usage optimal pour chaque approche

**Quand privilégier le SIEM :** détection d'attaques distribuées, corrélation multi-sources, reporting de conformité, vue globale de la sécurité.

**Quand privilégier l'EDR :** investigation post-incident, analyse de malware, détection de techniques avancées, réponse au niveau endpoint.

**Approche optimale :** combiner SIEM + EDR (XDR) pour bénéficier à la fois de la **largeur** (patterns globaux) et de la **profondeur** (détails techniques) de la visibilité.

## 3.IAM/PAM ET GESTION DES ACCÈS

La gestion des identités et des accès constitue un pilier fondamental de la sécurité des systèmes d'information. Cette section analyse comment notre lab Wazuh surveille et détecte les activités liées aux comptes utilisateurs et aux priviléges.

### 3.1 Identity and Access Management (IAM)

**IAM dans le lab :** gestion complète du cycle de vie des identités (création, modification, suppression, permissions, groupes).

**Événements surveillés :**

- **Authentifications** : SSH (Linux) et RDP (Windows, Event IDs 4624/4625) avec distinction des types de logon.
- **Gestion des comptes** : création (4720), modification (4738), suppression (4726), activation/désactivation (4722/4725).
- **Gestion des groupes** : ajout (4732) et retrait (4733), avec surveillance particulière des groupes **Administrators** et **Domain Admins**.

### 3.2 Privileged Access Management (PAM)

Le PAM se concentre spécifiquement sur les accès privilégiés (administrateurs, root, comptes de service). Ces comptes sont des cibles privilégiées des attaquants car ils offrent un contrôle total du système.

#### Surveillance PAM dans le lab :

##### Linux

- Commandes sudo : Rule 5402 trace toutes les élévations
- Connexions root directes : Alertées avec niveau High
- Modifications /etc/sudoers : Déetectées via FIM

##### Windows

- Utilisation de comptes Administrators : Event 4672 (Special privileges assigned)
- Élévation UAC : Visible dans Sysmon Event ID 1 (Integrity Level)
- RunAs : Déetectable via process parent-child relationships

### 3.3 Principes de sécurité IAM/PAM

**Principe du moindre privilège** : chaque utilisateur/processus n'a que les permissions nécessaires, surveillé via alertes sur comptes admin ou groupes privilégiés.

**Séparation des tâches (Separation of Duties)** : les admins disposent de comptes distincts pour usage quotidien et tâches administratives.

**Traçabilité complète** : toutes les actions privilégiées sont enregistrées et auditables, comme démontré avec Wazuh.

**Just-in-Time Access** : les accès admin doivent être temporaires et révoqués rapidement, via des solutions PAM modernes.

## 4. THREAT HUNTING ET DÉTECTION AVANCÉE

Le threat hunting représente une approche proactive de la cybersécurité, où les analystes recherchent activement des indicateurs de compromission plutôt que d'attendre passivement les alertes automatiques. Cette section présente plusieurs requêtes et méthodologies de hunting applicables à notre lab.

### 4.1 Concept de Threat Hunting

Le threat hunting est la pratique de rechercher proactivement des menaces qui ont échappé aux défenses existantes. Contrairement à la détection traditionnelle basée sur règles (signatures,

IOCs connus), le hunting suppose qu'un attaquant non détecté est déjà présent dans l'infrastructure et cherche des preuves de sa présence via analyse comportementale et anomalies statistiques.

### Méthodologies de hunting :

- **Hypothesis-driven** : Partir d'une hypothèse d'attaque et chercher les preuves
- **IOC-driven** : Rechercher des indicateurs de compromission connus
- **Analytics-driven** : Analyser des anomalies statistiques dans les données

## 4.2 Requêtes de Threat Hunting

### *Requête 1 : PowerShell suspect*

**Hypothèse** : Un attaquant utilise PowerShell avec des commandes encodées ou des téléchargements pour exécuter du code malveillant.

```
agent.name: "Windows-Client" AND  
data.win.system.eventID: "1" AND  
data.win.eventdata.image: "*powershell.exe" AND  
(data.win.eventdata.commandLine: "*IEX*" OR  
data.win.eventdata.commandLine: "*Invoke-Expression*" OR  
data.win.eventdata.commandLine: "*downloadstring*" OR  
data.win.eventdata.commandLine: "*EncodedCommand*")
```

**Analyse** : Cette requête détecte l'usage de techniques courantes d'attaquants via PowerShell. IEX (Invoke-Expression) exécute du code dynamiquement. DownloadString télécharge du code depuis Internet. EncodedCommand cache les commandes en base64. Ces patterns sont typiques de frameworks d'attaque comme Empire ou Cobalt Strike.

### *Requête 2 : Connexions réseau inhabituelles*

**Hypothèse** : Un malware établit une connexion C2 (Command and Control) sur un port non standard.

```
agent.name: "Windows-Client" AND
```

```
data.win.system.eventID: "3" AND
```

```
NOT data.win.eventdata.destinationPort: ("80" OR "443" OR "53" OR "445" OR "3389")
```

**Analyse :** Les ports 80/443 (HTTP/HTTPS), 53 (DNS), 445 (SMB), et 3389 (RDP) sont légitimes. Les connexions vers d'autres ports peuvent indiquer des backdoors ou C2. Le port 4444 est notamment associé à Metasploit. Cette requête identifie rapidement les anomalies réseau.

### ***Requête 3 : Modifications système critiques***

**Hypothèse :** Un attaquant modifie des fichiers système pour établir sa persistence ou compromettre le système.

rule.groups: "syscheck" AND

(syscheck.path: "/etc/\*" OR syscheck.path: "/bin/\*" OR syscheck.path: "/sbin/\*") AND  
syscheck.event: "modified"

**Analyse :** Les modifications de répertoires système critiques sont rares en production normale. /etc contient les configurations, /bin et /sbin les binaires système. Toute modification non planifiée peut indiquer une compromission ou l'installation d'un rootkit.

### **4.3 MITRE ATT&CK Mapping**

Les techniques détectées dans notre lab correspondent au framework MITRE ATT&CK :

Technique MITRE	ID	Détection dans le lab
Brute Force	T1110	SSH failures (Rule 5710), RDP failures (Event 4625)
PowerShell	T1059.001	Sysmon Event ID 1 avec powershell.exe
Create Account	T1136.001	Event 4720 + Rule 60106
Privilege Escalation	T1548	Sudo usage (Rule 5402), UAC bypass détectable
Valid Accounts	T1078.003	Monitoring authentications réussies
File and Directory Permissions Modification	T1222	FIM sur /etc/passwd
Remote Services: RDP	T1021.001	Event 4624/4625 Logon Type 10

## PARTIE V : RÉSULTATS ET CONCLUSION

### 1. RÉSULTATS OBTENUS

*Tableau récapitulatif des scénarios*

Scénario	Système	Type	Événements générés	Règles déclenchées	Détection	Sévérité	Technique MITRE
SSH Bruteforce	Linux	SIEM	10 tentatives auth	Rule 5710, 5503	✓ Immédiate	Medium (5)	T1110.001
Sudo élévation	Linux	PAM	3 commandes sudo	Rule 5402	✓ Immédiate	Low (3)	T1548
FIM /etc/passwd	Linux	SIEM	1 modification	Rule 550	✓ Immédiate	High (7)	T1222
RDP Failed	Windows	SIEM	5 tentatives RDP	Event 4625, Rule 60122	✓ Immédiate	Medium (5)	T1110.001
User Creation	Windows	IAM	2 events (4720, 4732)	Rules 60106, 60132	✓ Corrélés	Critical (9)	T1136.001
Sysmon Process	Windows	EDR	150+ événements	Sysmon ID 1	✓ Continue	Info (3)	T1059
Sysmon File	Windows	EDR	3 créations fichiers	Sysmon ID 11	✓ Continue	Info (3)	T1105
Sysmon Network	Windows	EDR	10+ connexions	Sysmon ID 3	✓ Continue	Info (3)	T1071
PowerShell Suspect	Windows	EDR	1 commande encodée	Custom Rule	✓ Immédiate	Critical (9)	T1059.001, T1140

*Métriques de performance :*

- Taux de détection :** 100% des scénarios testés ont été détectés
- Temps de détection :** < 30 secondes en moyenne (temps réel)
- Faux positifs :** 0 (dans le contexte du lab contrôlé)

- Couverture MITRE ATT&CK : 9 techniques détectées sur 14 tactics testées

### *Comparaison avec solutions commerciales*

Critère	Wazuh (Open-Source)	Splunk Enterprise Security	Elastic SIEM	Microsoft Sentinel
Coût licence	Gratuit	\$2,000/GB/an	\$95/GB/mois	\$2.46/GB/jour
Déploiement	Self-hosted	Cloud/On-prem	Cloud/On-prem	Cloud uniquement
Complexité	Moyenne-Élevée	Élevée	Moyenne	Moyenne
Communauté	Active	Très active	Très active	Microsoft ecosystem
EDR intégré	Via Sysmon	Non	Non	Via Defender
ML détection	Basique	Avancé	Avancé	Avancé
Threat Intel	Intégrations	Natif	Natif	Natif

### *Défis rencontrés et solutions*

Défi	Solution appliquée
Indexer ne démarre pas (mémoire insuffisante)	Passage de t2.small à t3.medium (8 GB RAM)
Agent Windows ne se connecte pas	Vérification Security Group port 1514/TCP
Sysmon non collecté par Wazuh	Ajout manuel section <localfile> dans ossec.conf
Alertes noyées dans bruit	Filtrage par niveau de严重性 >= Medium
Certificat SSL auto-signé bloqué	Acceptation exception navigateur (lab uniquement)

## 2. CONCLUSION GÉNÉRALE

Ce projet de fin d'études a représenté une expérience formatrice enrichissante qui m'a permis de concevoir, déployer et valider une plateforme complète de supervision de sécurité combinant les approches SIEM et EDR sur une infrastructure Cloud AWS. À travers la mise en œuvre de Wazuh et l'intégration de Sysmon, j'ai démontré qu'il est possible de bâtir une solution robuste de détection de menaces sans recourir à des solutions commerciales coûteuses. L'architecture déployée, composée de trois instances EC2 dans un environnement AWS sécurisé, a permis de superviser efficacement des endpoints Linux et Windows, avec une détection réussie des neuf scénarios d'attaque simulés en temps réel. Au-delà des aspects techniques, ce projet m'a permis d'acquérir une expertise approfondie en administration Cloud, analyse de logs complexes, threat hunting et compréhension du framework MITRE ATT&CK - des compétences directement valorisables en environnement SOC professionnel. J'ai particulièrement apprécié la complémentarité entre la vision macroscopique du SIEM et la vision microscopique de l'EDR, qui s'est révélée essentielle pour une défense en profondeur efficace. Bien que plusieurs axes d'amélioration aient été identifiés pour un déploiement en production (haute disponibilité, Threat Intelligence, automatisation SOAR), ce travail confirme que les solutions open-source comme Wazuh offrent une alternative crédible aux plateformes commerciales. Dans un contexte où les cyberattaques augmentent de 400% selon l'ANSSI et où le coût moyen d'une violation atteint 4,45 millions de dollars, ce projet m'a convaincu que la surveillance continue et la détection proactive sont vitales pour toute organisation moderne. Cette expérience a renforcé ma passion pour la cybersécurité défensive et m'a préparé à relever les défis d'un futur métier d'analyste SOC ou d'ingénieur en détection de menaces, où la vigilance constante, la formation continue et l'adaptation aux évolutions du paysage cyber seront mes priorités quotidiennes.