



Département Mathématiques et Informatique

Filière(s) : BDCC II

Groupe B

Atelier : Sécurité des Endpoints et Supervision SIEM – Étude de Cas Multi-OS (Linux & Windows)

Réalisé par :

NINEFLAS Laila

Supervisé par :

Prof.Azeddine KHIAT

Année Universitaire : 2025/2026

Table des matières

Introduction.....	3
Objectifs de l'atelier.....	3
1. Architecture du laboratoire AWS.....	4
2. Installation et configuration de Wazuh All-in-One.....	5
3. Configuration des communications et ports.....	7
4. Démonstration SIEM & EDR – Scénarios pratiques.....	8
4.1 Scénarios Linux.....	8
4.2 Scénarios Windows.....	11
Conclusion.....	14
Ressources GitHub.....	14

Introduction

Dans cet atelier pratique, nous avons mis en œuvre une plateforme de supervision et de protection de la sécurité basée sur **Wazuh**, intégrant **SIEM** (Security Information and Event Management) et **EDR** (Endpoint Detection and Response).

L'objectif est de fournir une expérience opérationnelle d'un **SOC (Security Operations Center)** moderne, permettant de détecter, analyser et répondre aux menaces dans un environnement Cloud multi-OS.

La plateforme repose sur AWS Learner Lab, avec :

- **Serveur Wazuh** : Ubuntu 22.04 LTS – collecte, corrélation, analyse et visualisation des événements
- **Client Linux** : Ubuntu 22.04 LTS – supervisé par l'agent Wazuh
- **Client Windows Server 2025** – supervisé par l'agent Wazuh, avec option Sysmon pour enrichir les événements EDR
- **VPC AWS** : réseau sécurisé avec règles de communication contrôlées entre les composants

Lien github : <https://github.com/NinflasLeila/endpoints-et-supervision-SIEM->

Objectifs de l'atelier

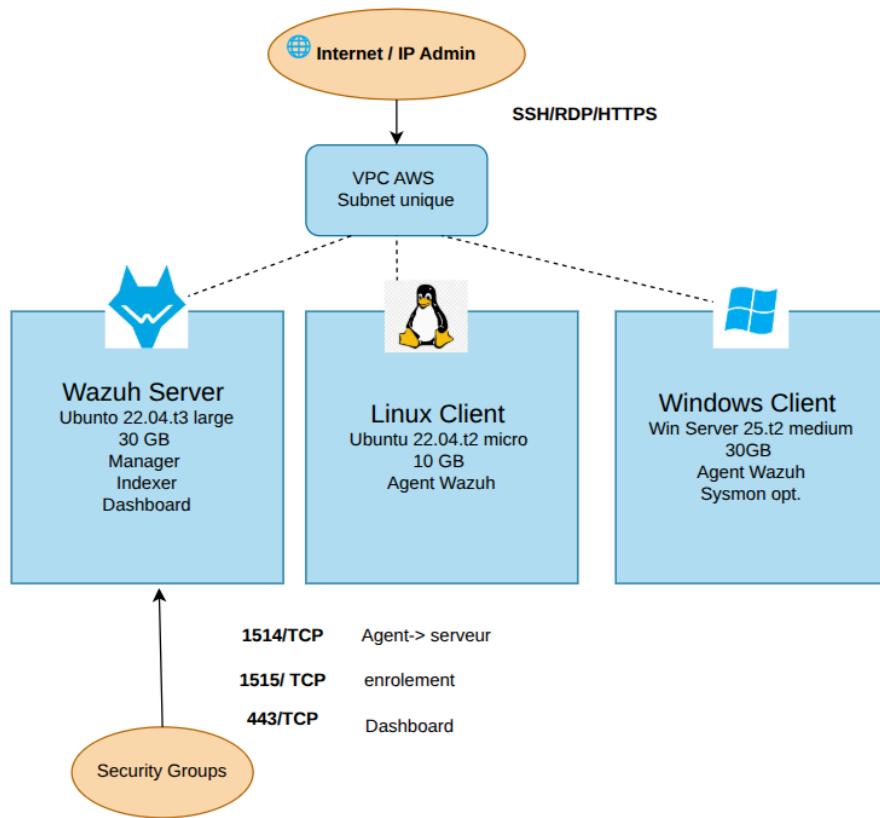
L'objectif de cet atelier est de comprendre l'architecture et le rôle d'un serveur SIEM, en mettant l'accent sur la gestion des endpoints. Il inclut l'installation et la configuration de Wazuh sur un serveur Ubuntu, ainsi que le déploiement et l'enrôlement d'agents sur des systèmes Linux et Windows. Les participants apprendront à générer des événements de sécurité réalistes et à les analyser, tout en explorant des aspects avancés tels que la corrélation d'événements, le *threat hunting* et la gestion des identités et des accès (*Identity & Access Management – IAM*).

1. Architecture du laboratoire AWS

Le laboratoire AWS est conçu autour de trois instances EC2, chacune ayant un rôle précis. Le **Wazuh-Server**, exécutant Ubuntu 22.04 LTS, joue le rôle de serveur Wazuh, d'indexeur et de tableau de bord, et est recommandé en type t3.large avec 30 Go de stockage.

Le **Linux-Client**, également sous Ubuntu 22.04 LTS, fonctionne comme agent Wazuh avec un type t2.micro ou t3.micro et 8 à 10 Go de stockage, tandis que le **Windows-Client**, équipé de Windows Server 2025, agit comme agent Wazuh (avec Sysmon optionnel) sur un type t2.medium et 30 Go de stockage. Les flux de communication sont définis par différents ports : le SSH (22/TCP) pour le Linux Client, le RDP (3389/TCP) pour le Windows Client, le HTTPS (443/TCP) pour accéder au dashboard Wazuh, et les ports 1514/TCP et 1515/TCP pour la communication et l'enrôlement des agents vers le serveur, avec le port 55000/TCP optionnel pour l'enrôlement via API.

L'architecture réseau repose sur un VPC unique avec un subnet unique, connecté à Internet pour SSH, RDP et HTTPS. Les instances sont organisées comme suit : le Wazuh-Server, le Linux-Client et le Windows-Client. Les **Security Groups** sont configurés de manière stricte : le SG du Wazuh-Server autorise les ports 22, 443, 1514 et 1515 uniquement depuis les clients et l'IP administrateur ; le SG du Linux-Client n'autorise que le port 22 depuis l'IP administrateur ; et le SG du Windows-Client n'autorise que le port 3389 depuis l'IP administrateur.



Après la création des instances de serveur WAZUH et deux agents :

Instances (3) Informations		Date de la dernière mise à jour	Se connecter	État de l'instance	Actions	Lancer des instances
		Il y a less than a minute		Tous les états		
<input type="checkbox"/>	Name	ID d'instance	État de l'insta...	Type d'insta...	Contrôle des statu	Statut d'alarme
<input type="checkbox"/>	Windows-Client	i-0499acc01bcc3d6c4		t3.micro	-	Afficher les alarm
<input type="checkbox"/>	Linux-Client	i-06d84abc30caae54a		t3.micro		Afficher les alarm
<input type="checkbox"/>	Wazuh-Server	i-02503b82afa2d3f21		t3.medium		Afficher les alarm

2. Installation et configuration de Wazuh All-in-One

- **Mise à jour du serveur et Installation Wazuh**

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

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

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

● Vérification des services

```
ubuntu@ip-172-31-28-52:~
```

```
Jan 08 11:50:49 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.entr>
Jan 08 11:50:49 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: System::setSecurityManager will be removed in a future release
Jan 08 11:50:51 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: A terminally deprecated method in java.lang.System has been called
Jan 08 11:50:51 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.Secur>
Jan 08 11:50:51 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.>
Jan 08 11:50:51 ip-172-31-28-52 systemd-entrypoint[15128]: WARNING: System::setSecurityManager will be removed in a future release
Jan 08 11:51:06 ip-172-31-28-52 systemd[1]: Started wazuh-indexer.service - Wazuh-Indexer.

ubuntu@ip-172-31-28-52:~
```

```
ubuntu@ip-172-31-28-52:~
```

```
ubuntu@ip-172-31-28-52:~
```

```
ubuntu@ip-172-31-28-52:~
```

```
● wazuh-dashboard.service - wazuh-dashboard
   Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; preset: enabled)
     Active: active (running) since Thu 2026-01-08 11:53:55 UTC; 1min 47s ago
       Main PID: 60852 (node)
         Tasks: 11 (limit: 4515)
        Memory: 187.7M (peak: 275.1M)
        CPU: 11.423s
      CGroup: /system.slice/wazuh-dashboard.service
             └─60852 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-rejections=warn /usr/share/wazuh-dashbo>
```

```
Jan 08 11:54:07 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:07Z", "tags": ["info", "plugins-service"]}>
Jan 08 11:54:07 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:07Z", "tags": ["info", "plugins-service"]}>
Jan 08 11:54:07 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:07Z", "tags": ["info", "plugins-service"]}>
Jan 08 11:54:07 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:07Z", "tags": ["info", "plugins-system"]}>
Jan 08 11:54:08 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:08Z", "tags": ["info", "savedobjects-ser>
Jan 08 11:54:08 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:08Z", "tags": ["info", "savedobjects-ser>
Jan 08 11:54:08 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:08Z", "tags": ["info", "plugins-system"]}>
Jan 08 11:54:09 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:09Z", "tags": ["listening", "info"], "pid":>
Jan 08 11:54:09 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "log", "@timestamp": "2026-01-08T11:54:09Z", "tags": ["info", "http", "server", ">
Jan 08 11:54:25 ip-172-31-28-52 opensearch-dashboards[60852]: {"type": "response", "@timestamp": "2026-01-08T11:54:24Z", "tags": [], "pid": 60852, "meth>
```

```
ubuntu@ip-172-31-28-52:~
```

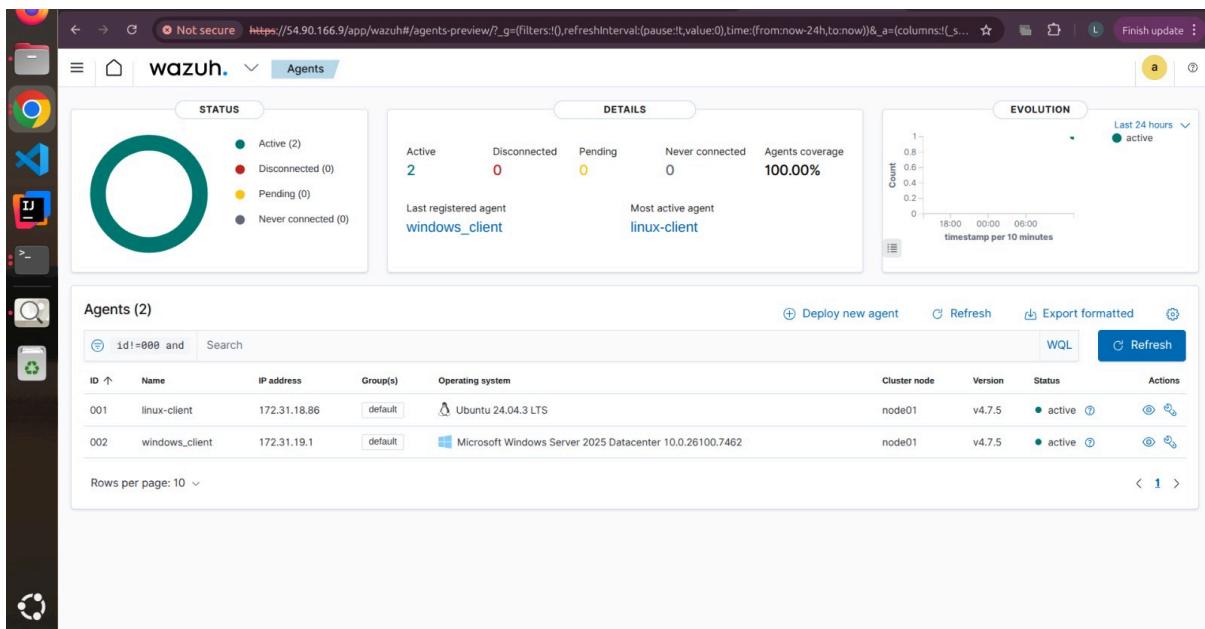
```
ubuntu@ip-172-31-28-52:~
```

```
ubuntu@ip-172-31-28-52:~
```

Interface WAZUH dans le browser :

- **Enrôlement des agents** : La création et l' enrôlement des deux agents sur WAZUH server

L'enrôlement des agents Wazuh se fait depuis le **Dashboard** via la section *Agents Management*, en suivant *Summary* → *Deploy new agent*. Pour l'**agent Linux**, il suffit de sélectionner *Linux*, puis de copier et exécuter les commandes proposées directement sur le Linux-Client. Pour l'**agent Windows**, on choisit *Windows* et on exécute la commande PowerShell sur le Windows-Client. Une fois l'installation terminée, il est important de vérifier que le service **Wazuh Agent** est bien en état *Running*. En option, l'installation de **Sysmon** peut être réalisée afin d'enrichir les journaux et améliorer la visibilité EDR.



3. Configuration des communications et ports

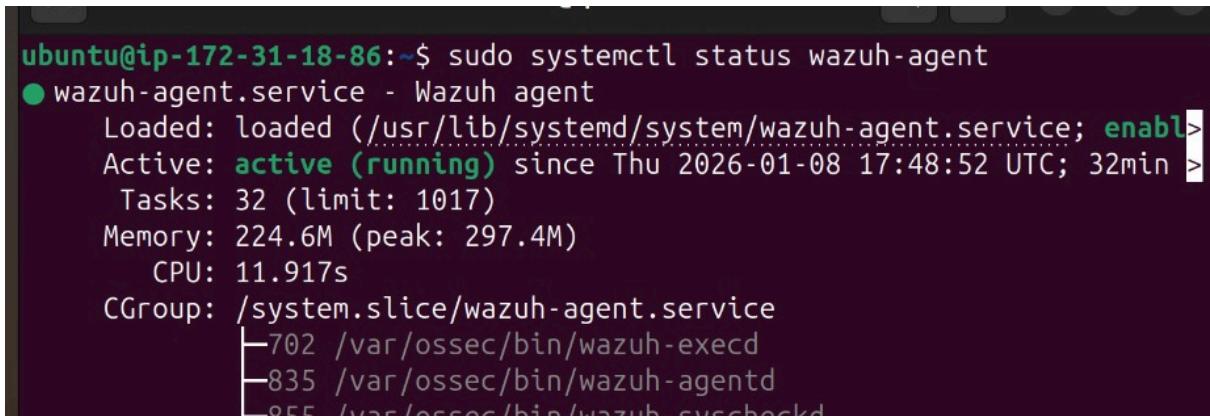
La configuration des communications repose sur l'ouverture des ports nécessaires dans les **Security Groups**. Les ports **1514** et **1515** sont autorisés pour la communication et l'enrôlement des agents, tandis que le tableau de bord Wazuh est accessible via **HTTPS** sur le port **443** (ou **5601** selon le type d'installation). Il est essentiel de vérifier que les agents communiquent correctement avec le serveur Wazuh pour assurer le bon fonctionnement de la supervision.

- Ports autorisés dans SG : 1514 + 1515 (communication + enrollment)

- Dashboard : HTTPS 443 (ou 5601 selon installation)
- Vérifier que les agents communiquent correctement avec le serveur Wazuh

4. Démonstration SIEM & EDR – Scénarios pratiques

4.1 Scénarios Linux



```
ubuntu@ip-172-31-18-86:~$ sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
  Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled)
  Active: active (running) since Thu 2026-01-08 17:48:52 UTC; 32min ago
    Tasks: 32 (limit: 1017)
   Memory: 224.6M (peak: 297.4M)
      CPU: 11.917s
     CGroup: /system.slice/wazuh-agent.service
             └─702 /var/ossec/bin/wazuh-execd
                 ├─835 /var/ossec/bin/wazuh-agentd
                 ├─855 /var/ossec/bin/wazuh-cvscheckd
```

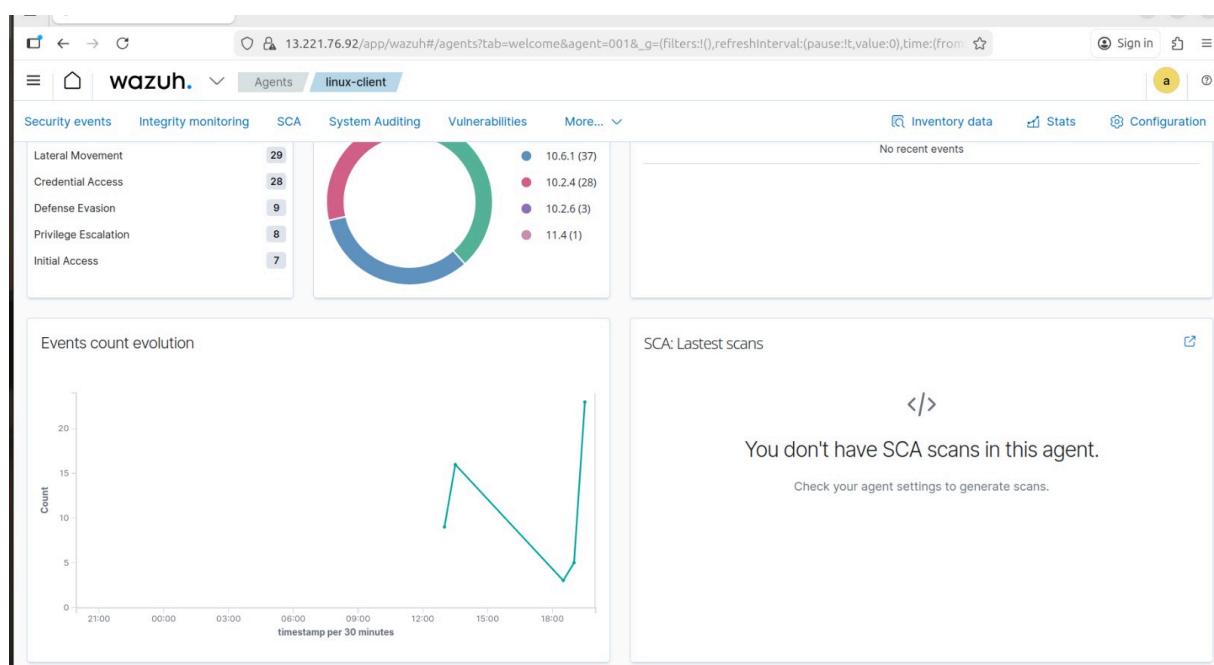
- **Tentatives SSH échouées**

Rôle / Purpose : Simuler des attaques de type brute force sur les accès SSH.

Exemple : `ssh fakeuser@IP_LINUX_CLIENT` répété 5 à 10 fois.

```
laila-nineflas@laila-nineflas:~/Downloads$ ssh -i "Key_3.pem" ubuntas@98.81.158.102
ubuntas@98.81.158.102: Permission denied (publickey).
laila-nineflas@laila-nineflas:~/Downloads$ ssh -i "Key_3.pem" ubuntas@98.81.158.102
ubuntas@98.81.158.102: Permission denied (publickey).
laila-nineflas@laila-nineflas:~/Downloads$ ssh -i "Key_3.pem" ubuntas@98.81.158.102
ubuntas@98.81.158.102: Permission denied (publickey).
laila-nineflas@laila-nineflas:~/Downloads$ ssh -i "Key_3.pem" ubuntas@98.81.158.102
ubuntas@98.81.158.102: Permission denied (publickey).
laila-nineflas@laila-nineflas:~/Downloads$
```

Résultat : alertes “authentication failed” / “sshd”



Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 8, 2026 @ 19:56:24.488	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:22.486	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:18.483	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:14.479	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:08.472	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:06.470	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:02.467	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:56:00.465	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:55:56.461	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710
> Jan 8, 2026 @ 19:55:52.457	T1110.001	T1021.004	sshd: Attempt to login using a non-existent user	5	5710

- **Élévation de priviléges**

Rôle / Purpose : Vérifier la détection des actions suspectes de priviléges administratifs.

Exemple : `sudo su` sur le Linux-Client.

```
ubuntu@ip-172-31-18-86:~$ sudo su
root@ip-172-31-18-86:/home/ubuntu# exit
exit
ubuntu@ip-172-31-18-86:~$ sudo whoami
root
ubuntu@ip-172-31-18-86:~$ sudo -u autreuser ls
sudo: unknown user autreuser
sudo: error initializing audit plugin sudoers_audit
ubuntu@ip-172-31-18-86:~$ 
```

Résultat : événements sudo

Security Alerts					
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 8, 2026 @ 20:03:34.895			PAM: Login session closed.	3	5502
> Jan 8, 2026 @ 20:03:34.893	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
> Jan 8, 2026 @ 20:03:34.892	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
> Jan 8, 2026 @ 20:03:26.885			PAM: Login session closed.	3	5502

- **Modification fichier sensible (FIM)**

Rôle / Purpose : Tester le système de surveillance d'intégrité des fichiers (File Integrity Monitoring).

Exemple : `echo "test" | sudo tee -a /etc/passwd`.

```
ubuntu@ip-172-31-18-86:~$ echo "test" | sudo tee -a /etc/passwd
test
ubuntu@ip-172-31-18-86:~$
```

Résultat : alerte File Integrity Monitoring

Security Alerts					
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 8, 2026 @ 20:03:34.895			PAM: Login session closed.	3	5502
> Jan 8, 2026 @ 20:03:34.893	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
> Jan 8, 2026 @ 20:03:34.892	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
> Jan 8, 2026 @ 20:03:26.885			PAM: Login session closed.	3	5502

4.2 Scénarios Windows

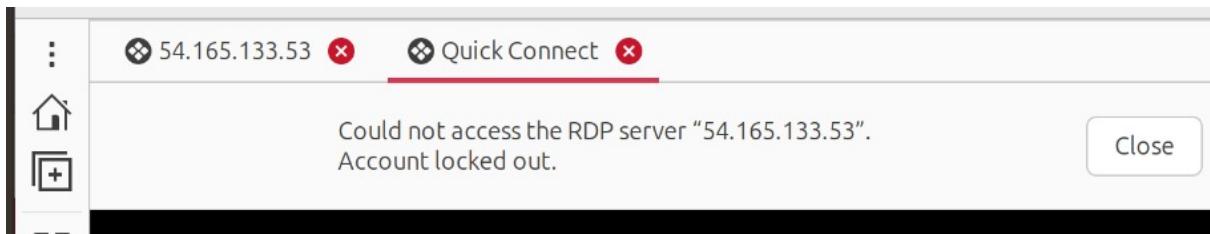
1. Échecs de login RDP

Rôle / Purpose : Simuler des tentatives de connexion non autorisées via RDP pour tester la détection d'accès frauduleux.

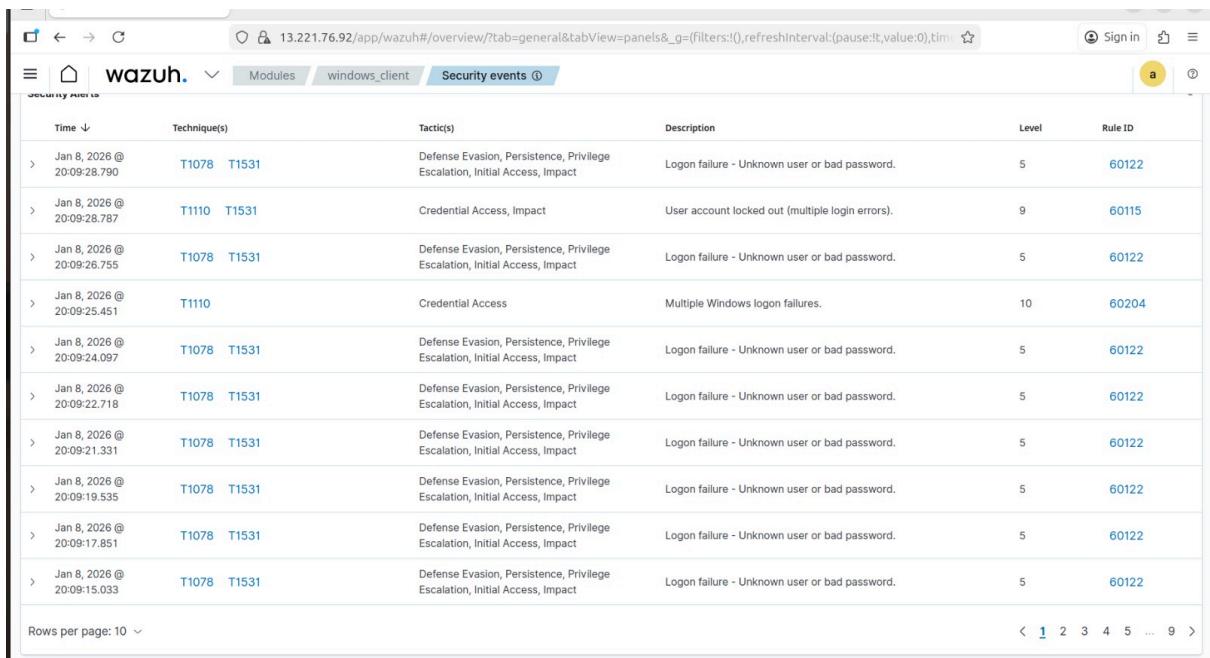
Exemple : Connexions RDP avec des mauvais mots de passe 2 à 5 fois.

Résultat attendu : Génération d'événements Windows Security (4625).

- Connexions RDP avec mauvais mot de passe (2–5 fois)



Résultat : événements Windows Security (4625)

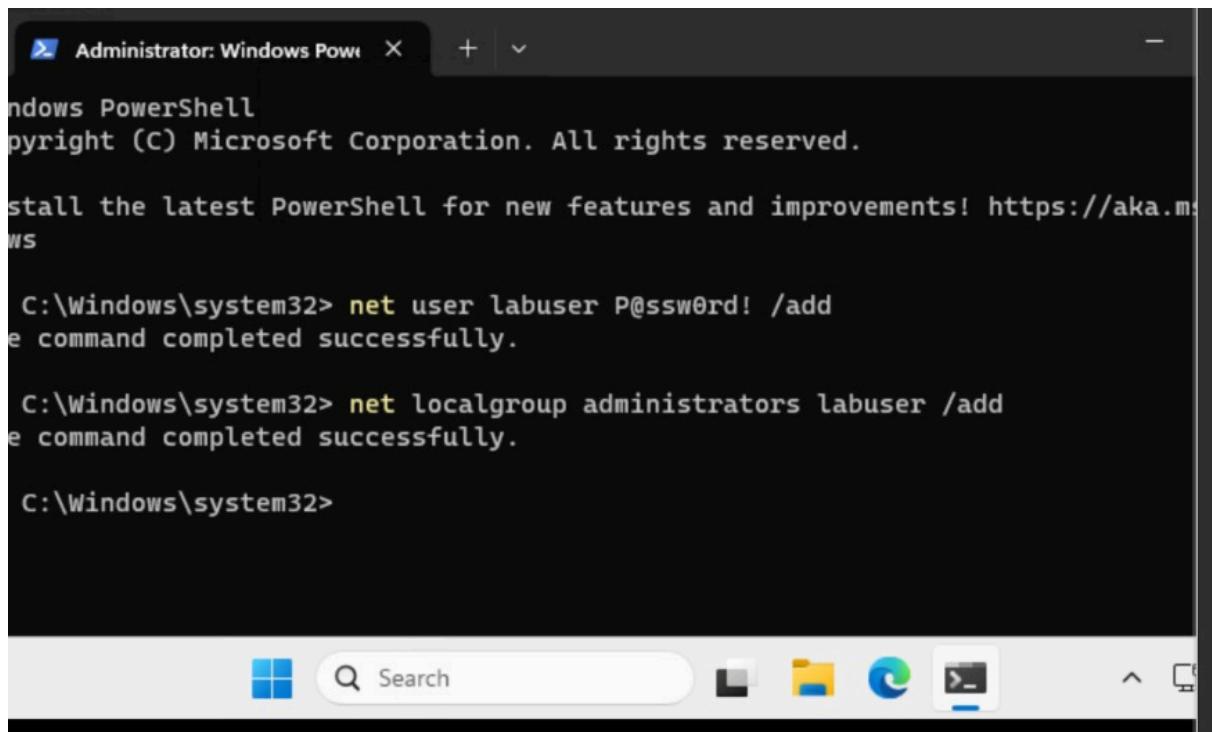


Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 8, 2026 @ 20:09:28.790	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:28.787	T1110 T1531	Credential Access, Impact	User account locked out (multiple login errors).	9	60115
Jan 8, 2026 @ 20:09:26.755	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:25.451	T1110	Credential Access	Multiple Windows logon failures.	10	60204
Jan 8, 2026 @ 20:09:24.097	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:22.718	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:21.331	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:19.535	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:17.851	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122
Jan 8, 2026 @ 20:09:15.033	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.	5	60122

2. Création d'utilisateur local

Rôle / Purpose : Vérifier la surveillance des changements dans la configuration des comptes utilisateurs et groupes.

Exemple :



```

Administrator: Windows PowerShell X + v

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

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

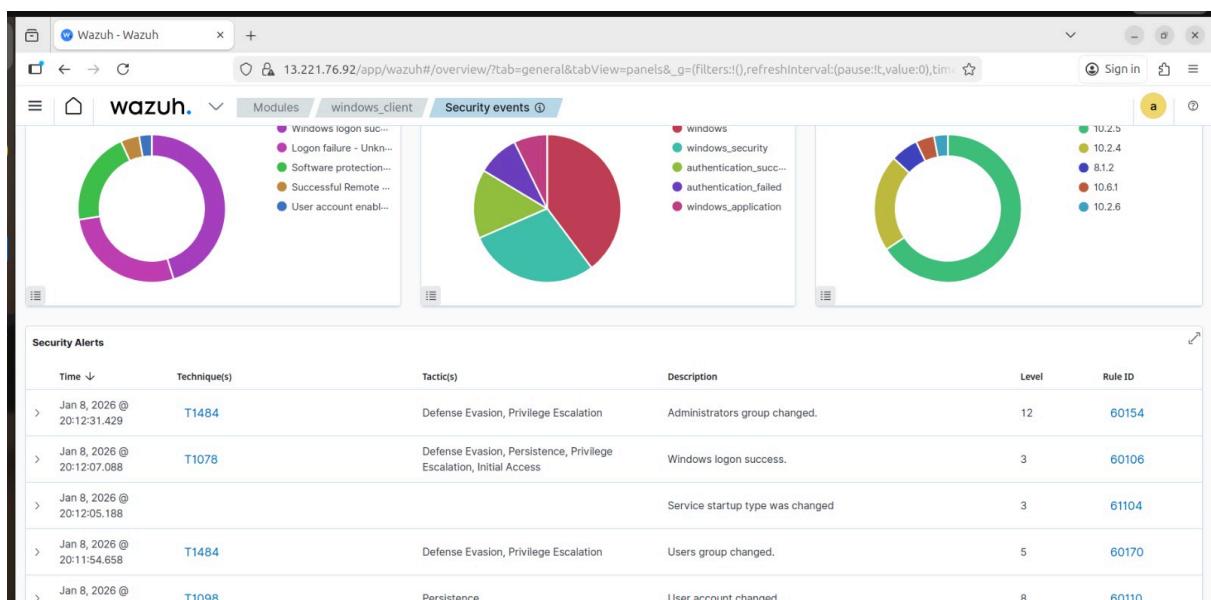
C:\Windows\system32> net user labuser P@ssw0rd! /add
The command completed successfully.

C:\Windows\system32> net localgroup administrators labuser /add
The command completed successfully.

C:\Windows\system32>

```

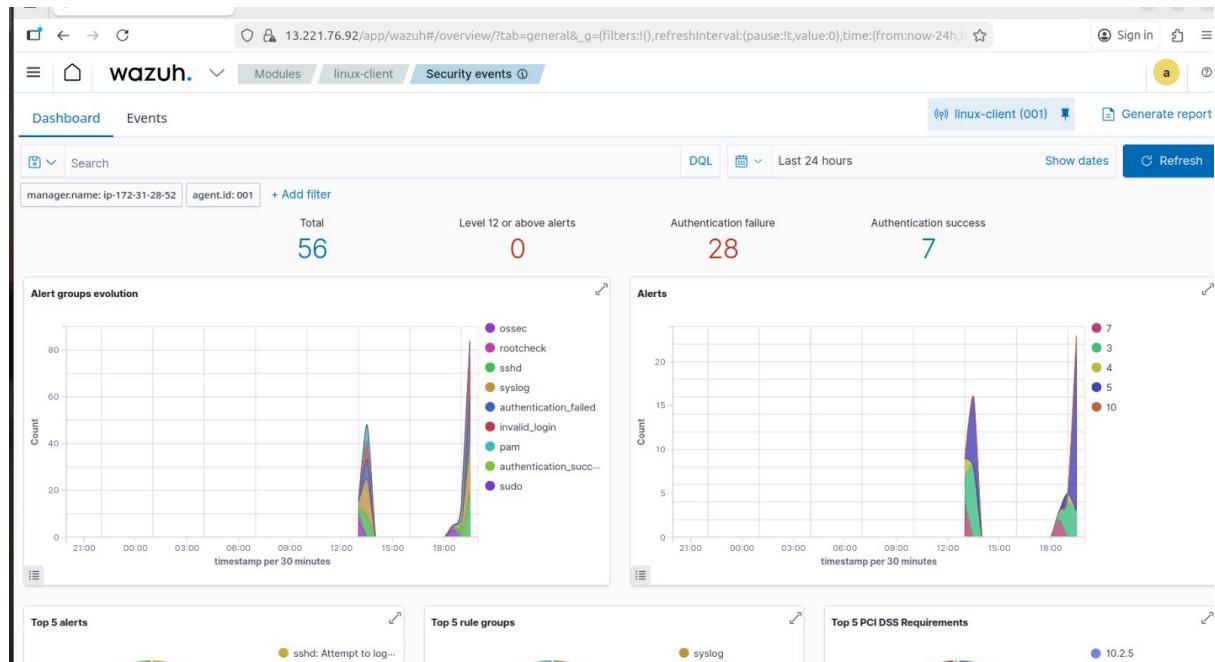
Résultat : événements “user created / group changed”



3. Option Sysmon : log des process et connexions réseau pour démonstration EDR avancée

Rôle / Purpose : Enrichir la visibilité EDR avec le logging des processus et des connexions réseau pour des démonstrations avancées.

Résultat attendu : Journaux détaillés pour le threat hunting et la corrélation d'événements.



Conclusion

En conclusion, cet atelier pratique a permis de mettre en œuvre une plateforme complète de supervision et de protection des endpoints à l'aide de Wazuh, en combinant les approches SIEM et EDR. Grâce à l'installation des agents sur des systèmes Linux et Windows, à la génération d'événements de sécurité réalistes, et à l'analyse des alertes, il a été possible d'illustrer la détection de tentatives d'intrusion, d'élevations de priviléges, de modifications de fichiers sensibles et de changements dans les comptes utilisateurs. L'exploration des fonctionnalités avancées telles que la corrélation d'événements, le *threat hunting* et la gestion des identités (*IAM*) a démontré l'importance d'une supervision centralisée pour la sécurité d'un système multi-OS. Cet atelier souligne ainsi la valeur stratégique des solutions SIEM/EDR pour prévenir, détecter et répondre efficacement aux menaces sur un réseau d'entreprise.

Ressources GitHub

- Documentation officielle Wazuh : <https://documentation.wazuh.com>
- Dépôt GitHub du :

<https://github.com/NinflasLeila/endpoints-et-supervision-SIEM->