
Mise en place d'un système de monitoring SIEM

Réalisé par **Argoubi Ahmed**

Introduction Générale

Dans le domaine de la cybersécurité, les menaces sont en constante évolution et deviennent de plus en plus sophistiquées. Les entreprises font face à une multitude de défis, allant des attaques par phishing et des logiciels malveillants aux violations de données et aux ransomwares. Ces menaces peuvent entraîner des pertes financières considérables, des dommages à la réputation et des perturbations opérationnelles majeures.

Pour faire face à ces défis, les solutions SIEM (Security Information and Event Management) jouent un rôle essentiel. Un SIEM permet de centraliser la collecte, l'analyse et la corrélation des données de sécurité provenant de diverses sources au sein de l'organisation. En temps réel, il identifie et alerte sur les comportements anormaux et les incidents de sécurité potentiels. Grâce à sa capacité à analyser des volumes massifs de données et à détecter des schémas de menaces, un SIEM améliore considérablement la visibilité sur l'ensemble du réseau.

Ce travail sera divisé en deux chapitres.

- Le premier chapitre , va présenter la problématique du projet, une solution afin de résoudre le problème trouvé puis l'étude théorique de cette solution
- Le deuxième chapitre, va focaliser sur l'installation de l'environnement du travail et à la réalisation pratique la solution adoptée pour assurer la supervision.

Chapitre 1 : Etude théorique

1. Présentation générale du projet:

Face à l'augmentation constante des cybermenaces, il est devenu crucial pour les organisations de mettre en place des mesures de sécurité efficaces pour protéger leurs infrastructures informatiques. Une solution efficace pour renforcer la cybersécurité est l'implémentation d'un système de gestion des informations et des événements de sécurité (SIEM). Ce projet se propose de déployer un système de monitoring SIEM en utilisant la pile ELK (Elasticsearch, Logstash, Kibana) intégrée à Wazuh.

1.1 La problématique :

Dans un contexte où les cybermenaces se multiplient et deviennent de plus en plus sophistiquées, les entreprises peinent à détecter et à réagir efficacement aux incidents de sécurité. La fragmentation des systèmes de sécurité et la gestion de volumes massifs de données provenant de diverses sources compliquent la tâche des équipes IT. Il est donc crucial d'implémenter des mesures de sécurité efficaces pour protéger les systèmes.

1.2 Solution proposée :

Pour répondre à cette problématique, la mise en place d'un système de monitoring SIEM basé sur l'ELK Stack (Elasticsearch, Logstash et Kibana) s'avère être une solution efficace. L'ELK Stack permet de centraliser, d'analyser et de visualiser en temps réel les données de sécurité provenant de diverses sources.

Grâce à ses capacités avancées de corrélation et de visualisation des données, un SIEM offre une vue d'ensemble des activités suspectes et des menaces potentielles. Cela permet non seulement de réagir rapidement aux incidents de sécurité, mais aussi de garantir la conformité aux réglementations en vigueur et d'optimiser la gestion des ressources de l'organisation.

2. Centre opérationnel de sécurité (SOC) :

Le Security Operations Center (SOC) est une unité centrale au sein d'une organisation chargée de surveiller, détecter, analyser et répondre aux incidents de sécurité.



Figure 1 : Fonctionnement de SOC

Dans un Security Operations Center (SOC), le SIEM (Security Information and Event Management) est une composante essentielle. Le SIEM joue un rôle crucial en centralisant, analysant et corrélant les données de sécurité provenant de diverses sources au sein de l'organisation. Il permet au SOC de détecter rapidement les incidents de sécurité, de fournir une surveillance en temps réel, de générer des alertes et des rapports, et de faciliter une réponse rapide aux menaces. Ainsi, le SIEM est intégré dans l'architecture et les processus opérationnels d'un SOC pour renforcer la sécurité globale de l'entreprise.

3. SIEM :

Un SIEM (Security Information and Event Management) est un système qui centralise, analyse et corrèle les données de sécurité provenant de diverses sources pour détecter et répondre rapidement aux menaces. Il fournit une surveillance en temps réel, des alertes de sécurité et des rapports de conformité .

3.1 Fonctionnalités Principales d'un SIEM :

- **Collecte de Données** : Centralisation des journaux et des événements de sécurité provenant de divers points de terminaison et dispositifs réseau.
- **Corrélation des Événements** : Analyse et mise en relation des données collectées pour identifier des modèles d'attaques complexes et des menaces potentielles grâce à des règles de corrélation sophistiquées.
- **Détection des Menaces** : Utilisation de signatures connues et de techniques d'analyse comportementale pour identifier les comportements anormaux et les anomalies en temps réel.
- **Alertes et Notifications** : Génération et envoi d'alertes instantanées lorsqu'une menace ou une anomalie est détectée, via divers canaux comme les emails, SMS ou tableaux de bord.
- **Rapports et Tableaux de Bord** : Création de rapports détaillés et visualisation des données de sécurité à travers des tableaux de bord interactifs et personnalisables, aidant à la prise de décision et à la conformité réglementaire.

3.2 Les 2 Types de Sources de Journaux Réseaux :

Les sources de journaux réseau peuvent être divisées en deux catégories logiques :

3.2.1 Sources de Journaux Centrés sur l'Hôte :

Ces journaux capturent les événements survenus au sein ou en lien avec l'hôte, tels que les journaux d'événements Windows, Sysmon, et Osquery. Exemples de journaux centrés sur l'hôte:

- Accès à un fichier par un utilisateur
- Tentative d'authentification d'un utilisateur
- Exécution d'un processus
- Modification de clés ou valeurs de registre par un processus
- Exécution de PowerShell

3.2.2 Sources de Journaux Centrés sur le Réseau :

Ces journaux sont générés lorsque les hôtes communiquent entre eux ou accèdent à internet. Exemples de protocoles réseau : SSH, VPN, HTTP/s, FTP. Exemples d'événements réseau :

- Connexion SSH

- Accès à un fichier via FTP
- Trafic web
- Accès aux ressources de l'entreprise via VPN
- Partage de fichiers en réseau

4. Présentation de ELK et Wazuh :

4.1 ELK :

Elasticsearch, auparavant ELK Search, est un ensemble de solutions logicielles. (ELK est l'acronyme des composants Elasticsearch, Logstash et Kibana.)

- **Elasticsearch** : Un moteur de recherche et d'analyse distribué qui indexe et stocke les données de sécurité.
- **Logstash** : est un collecteur des logs qui offre la possibilité de collecter, de traiter et de normaliser les données.
- **Kibana** : Un outil de visualisation qui permet de créer des tableaux de bord interactifs pour analyser les données stockées dans Elasticsearch.

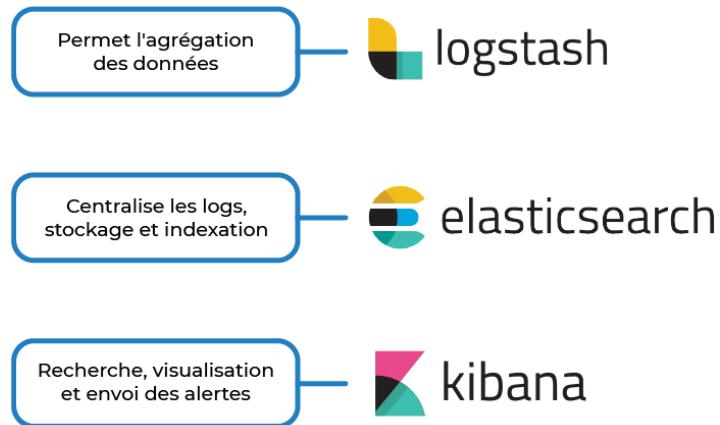


Figure 2 : Composants du ELK

4.2 Wazuh :

Un module de sécurité open-source qui s'intègre à ELK, enrichissant ses capacités avec des fonctionnalités avancées telles que la surveillance de l'intégrité des fichiers, la détection des rootkits, la surveillance des configurations de sécurité, et la gestion des alertes de sécurité.

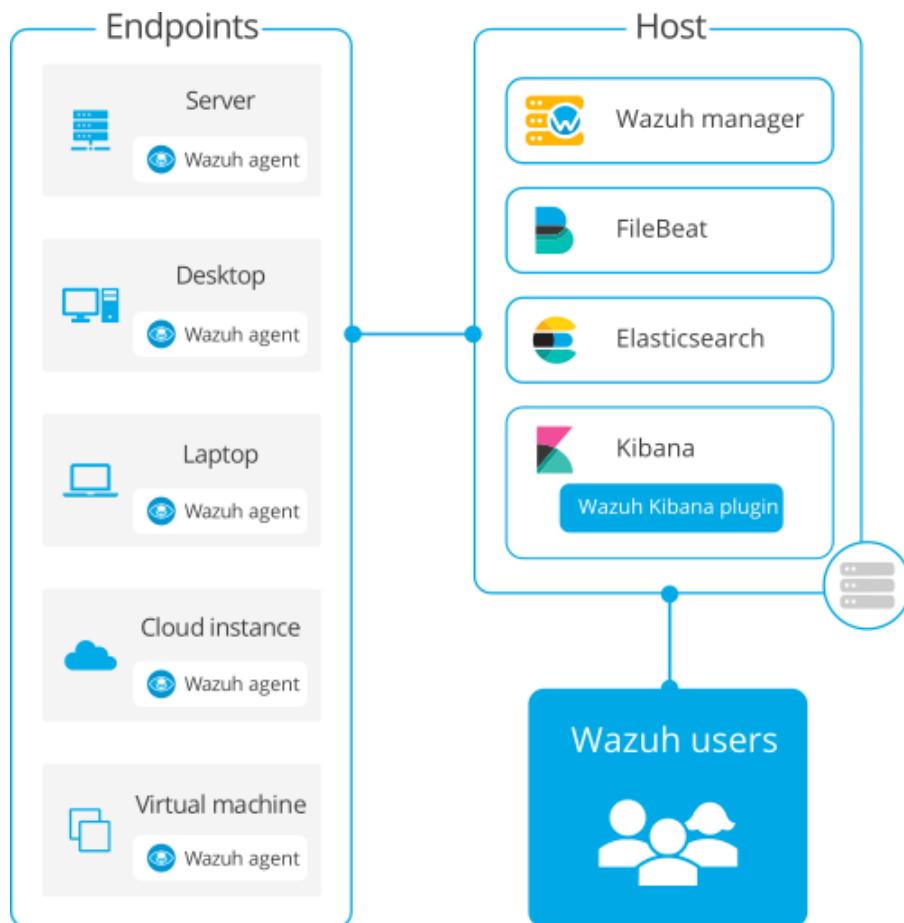


Figure 3 : Déploiement wazuh avec elk

Wazuh collecte les journaux de sécurité des systèmes et des applications et les envoie à Logstash pour un traitement initial. Logstash transforme les journaux en un format structuré et les envoie à Elasticsearch pour l'indexation. Elasticsearch stocke ces journaux indexés, facilitant une recherche rapide et une analyse en profondeur. Kibana offre des outils puissants pour visualiser les données indexées, permettant la création de tableaux de bord interactifs et la génération de rapports.

- Wazuh fournit des capacités supplémentaires telles que la détection d'intrusions, la gestion des vulnérabilités, et la conformité aux réglementations, enrichissant ainsi la pile ELK avec des fonctionnalités de sécurité spécifiques

Chapitre 2 Mise en place et réalisation

1. Introduction :

Dans ce dernier chapitre, je présenterons l'environnement du travail matériel, logiciel et l'architecture du travail. Ainsi que, ainsi les tâches réalisées .

2. Environnement de travail :

2.1 Environment matériel

Durant mon travail, j'ai eu recours à l'utilisation :

Laptop :



Manufacturer : HP (Victus)

Edition : Windows 11 Home

Processeur : AMD Ryzen 7 5000H

RAM : 32 GB

Disque Dur : SSD 512 GB

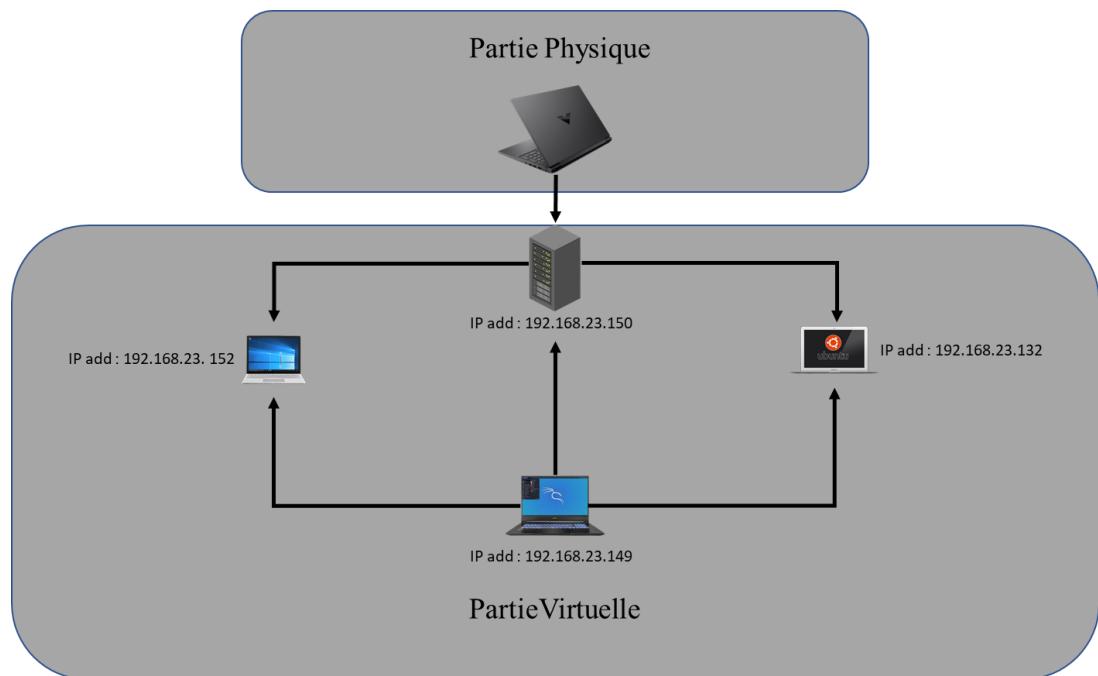
Sur l'ordinateur portable mentionné j'ai installé 4 machines virtuelles linux , 2 machines Ubuntu virtuelle , une machine Windows et une machine kali , ayant les caractéristiques décrites dans le tableau ci-dessous .

Tableau 1 : Les Machines virtuelle utiliser

	Machine virtuelle 1 : Serveur	Machine virtuelle 2 : Agent 1	Machine virtuelle 3 : Agent 2	Machine virtuelle 4 : Pirate
Système d'exploitation	ubuntu-22.04.3	ubuntu-22.04.3	Windows 10	Kali Linux
RAM	8 GB	6 GB	6 GB	8 GB
Disque Dur	100 GB	25 GB	40 GB	100 GB

L'architecture décrite dans la Figure ci-dessous représente un système SIEM avec une machine physique et un environnement virtualisé VMware. Ce système comprend une machine ubuntu-22.04.3 qui contient ELK et Wazuh agissant comme un SIEM qui connecté à deux agents , une machine Windows 10 et une autre machine Ubuntu 22.04.

Une machine Kali est également présente pour mener quelques types d'attaques contre les agents.



2.2 Environment logiciel :

- **VMware workstation** : VMware Workstation est un outil de virtualisation créé par la société VMware il peut être utilisé pour mettre en place un environnement de test pour développer de nouveaux logiciels, ou pour tester l'architecture complexe d'un système d'exploitation avant de l'installer réellement sur une machine physique. (1)



- **Kali Linux** : Kali Linux est une distribution Linux open source basée sur Debian et orientée vers diverses tâches de sécurité de l'information, telles que les tests d'intrusion, la recherche sur la sécurité, l'investigation informatique et l'ingénierie inverse .



- **Ubuntu** : Ubuntu est une distribution Linux fondée sur Debian. Elle est développée, commercialisée et maintenue pour les ordinateurs individuels, les serveurs et les objets connectés par la société Canonical



- **Hydra** : Hydra est un cracker de connexion réseau parallélisé intégré à divers systèmes d'exploitation comme Kali Linux , Parrot et d'autres environnements de test d'intrusion majeurs . Hydra fonctionne en utilisant différentes approches pour effectuer des attaques par force brute afin de deviner la bonne combinaison de nom d'utilisateur et de mot de passe .



- **Elastic stack** : ELK est un outil d'analyse de logs composé de 3 logiciels open source, développés par la société Elastic : Elasticsearch, Logstash et Kibana.



- **Wazuh** : La solution Wazuh Security Information and Event Management (SIEM) permet de surveiller, de détecter et d'alerter les événements et incidents de sécurité.

3. Installation ELK :

3.1 Installation d'Elasticsearch :

3.1.1 Ajout du Dépôt Elastic Stack :

Pour installer Elasticsearch, j'ai d'abord ajouté la clé GPG d'Elastic et l'ai importée dans un keyring sécurisé , ensuite, j'ai ajouté le dépôt Elastic Stack à la liste des sources APT .

```
root@ahmed:~# curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/elasticsearch.gpg --import && chmod 644 /usr/share/keyrings/elasticsearch.gpg
gpg: keyring '/usr/share/keyrings/elasticsearch.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key D27D666CD88E42B4: public key "Elasticsearch (Elasticsearch Signing Key) <dev_ops@elasticsearch.org>" imported
gpg: Total number processed: 1
gpg:           imported: 1
root@ahmed:~# echo "deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main
root@ahmed:~# apt-get update
Hit:1 http://tn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://tn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://tn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Reading package lists... Done
```

Ces étapes assurent que notre système est configuré pour installer Elasticsearch de manière sécurisée et efficace en utilisant les sources officielles maintenues par Elastic.

3.1.2 Installation et Configuration d'Elasticsearch:

Pour installer Elasticsearch, j'ai d'abord installé le package avec la version spécifiée (7.17.13) avec la commande :

✓ apt-get install elasticsearch=7.17.13

Puis, j'ai téléchargé le fichier de configuration pour Elasticsearch

```
root@ahmed:~# curl -so /etc/elasticsearch/elasticsearch.yml https://packages.wazuh.com/4.5/tpl/elastic-basic/elasticsearch_all_in_one.yml
root@ahmed:~#
```

3.1.3 Crédit et Déploiement des Certificats :

J'ai téléchargé le fichier de configuration pour créer les certificats. Ensuite, j'ai créé les certificats nécessaires à l'aide de l'outil **elasticsearch-certutil**

```
root@ahmed:~# curl -so /usr/share/elasticsearch/instances.yml https://packages.wazuh.com/4.5/tpl/elastic-basic/instances_aio.yml
root@ahmed:~# /usr/share/elasticsearch/bin/elasticsearch-certutil cert ca --pem --in instances.yml --keep-ca-key --out ~/certs.zip
This tool assists you in the generation of X.509 certificates and certificate
signing requests for use with SSL/TLS in the Elastic stack.

The 'cert' mode generates X.509 certificate and private keys.
* By default, this generates a single certificate and key for use
  on a single instance.
* The '-multiple' option will prompt you to enter details for multiple
  instances and will generate a certificate and key for each one
* The '-in' option allows for the certificate generation to be automated by describing
  the details of each instance in a YAML file

* An instance is any piece of the Elastic Stack that requires an SSL certificate.
  Depending on your configuration, Elasticsearch, Logstash, Kibana, and Beats
  may all require a certificate and private key.
* The minimum required value for each instance is a name. This can simply be the
  hostname, which will be used as the Common Name of the certificate. A full
  distinguished name may also be used.
* A filename value may be required for each instance. This is necessary when the
  name would result in an invalid file or directory name. The name provided here
  is used as the directory name (within the zip) and the prefix for the key and
```

J'ai extrait le fichier généré **certs.zip**. Puis, j'ai créé le répertoire **/etc/elasticsearch/certs** et y ai copié les fichiers CA, le certificat et la clé

```
root@ahmed:~# unzip ~/certs.zip -d ~/certs
Archive: /root/certs.zip
  creating: /root/certs/ca/
  inflating: /root/certs/ca/ca.crt
  inflating: /root/certs/ca/ca.key
  creating: /root/certs/elasticsearch/
  inflating: /root/certs/elasticsearch/elasticsearch.crt
  inflating: /root/certs/elasticsearch/elasticsearch.key
root@ahmed:~# mkdir /etc/elasticsearch/certs/ca -p
cp -R ~/certs/ca/ ~/certs/elasticsearch/* /etc/elasticsearch/certs/
chown -R elasticsearch: /etc/elasticsearch/certs
chmod -R 500 /etc/elasticsearch/certs
chmod 400 /etc/elasticsearch/certs/ca/ca.* /etc/elasticsearch/certs/elasticsearch.*
rm -rf ~/certs/ ~/certs.zip
```

Ensuite, j'ai activé et démarré le service Elasticsearch a l'aide de ces 3 commandes :

- ✓ systemctl daemon-reload
- ✓ systemctl enable elasticsearch
- ✓ systemctl start elasticsearch

Ensute , j'ai générée les informations d'identification de la Suite Elastic.

```
root@ahmed:~# /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.
The passwords will be randomly generated and printed to the console.
Please confirm that you would like to continue [y/N]

Changed password for user apm_system
PASSWORD apm_system = uWjrJSK5Y74kHqhTpnsL

Changed password for user kibana_system
PASSWORD kibana_system = pFIqR0x9stibxm9MpzXr

Changed password for user kibana
PASSWORD kibana = pFIqR0x9stibxm9MpzXr

Changed password for user logstash_system
PASSWORD logstash_system = z8IoQnyTQtqgrj8JpYFj

Changed password for user beats_system
PASSWORD beats_system = pYQ32PofYHit52inizb6

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = SBwsz0RrzGmVsM5lops0

Changed password for user elastic
PASSWORD elastic = V4rrqfBJo7Y0RhpmidHRk
```

3.1.4 Vérification de l'Installation :

Pour vérifier que l'installation s'est déroulée correctement, j'ai exécuté la commande suivante :

```
root@ahmed:~# curl -XGET https://localhost:9200 -u elastic:V4rrqfBJo7Y0RhpmidHRk -k
{
  "name" : "elasticsearch",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "UC433UMiSp6MLX2jygqjRA",
  "version" : {
    "number" : "7.17.13",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "2b211dbb8bfdecaf7f5b44d356bdfe54b1050c13",
    "build_date" : "2023-08-31T17:33:19.958690787Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
```

Cette commande a renvoyé une sortie confirmant la réussite de l'installation .

3.2 Installation du Serveur Wazuh :

Pour configurer Wazuh, la première étape consiste à ajouter le dépôt Wazuh au serveur.

3.2.1 Ajout du Dépôt Wazuh :

J'ai commencé par installer la clé GPG et ensuite, j'ai ajouté le dépôt Wazuh à la liste des sources APT

```
root@ahmed:~# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg: imported: 1
root@ahmed:~# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
root@ahmed:~# apt-get update
Hit:1 http://tn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://tn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:3 https://packages.wazuh.com/4.x/apt/stable InRelease [17.3 kB]
Hit:4 http://tn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 https://artifacts.elastic.co/packages/7.x/apt/stable InRelease
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:7 https://packages.wazuh.com/4.x/apt/stable/main amd64 Packages [40.8 kB]
Get:8 https://packages.wazuh.com/4.x/apt/stable/main i386 Packages [11.6 kB]
Fetched 69.6 kB in 1s (73.2 kB/s)
Reading package lists... Done
```

3.2.2 Installation du Manager Wazuh

J'ai installé le package Wazuh manager avec la commande :

- ✓ apt-get install wazuh-manager=4.5.4-1

Puis, j'ai activé et démarré le service Wazuh manager a l'aide de ces commandes :

- ✓ systemctl daemon-reload
- ✓ systemctl enable wazuh-manager
- ✓ systemctl start wazuh-manager

Pour vérifier que le manager Wazuh est actif, j'ai exécuté la commande suivante :

```
root@ahmed:~# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-06-01 08:00:20 CET; 6s ago
     Process: 48827 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
       Tasks: 120 (limit: 9386)
      Memory: 676.5M
        CPU: 21.790s
      CGroup: /system.slice/wazuh-manager.service
              └─48883 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
                  ├─48923 /var/ossec/bin/wazuh-authd
                  ├─48929 /var/ossec/bin/wazuh-db
```

3.3 Installation de Filebeat :

3.3.1 Installation et Configuration de Filebeat :

Pour installer Filebeat, j'ai d'abord installé le package Filebeat avec la commande suivante :

✓ apt-get install filebeat=7.17.13

Ensuite, j'ai téléchargé le fichier de configuration de Filebeat pour transférer les alertes Wazuh vers Elasticsearch à l'aide de commandes suivantes :

✓ curl -sO /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.5/tpl/elasticsearch/basic/filebeat_all_in_one.yml.

Et après le téléchargement du fichier de configuration de Filebeat et le modèle d'alertes pour Elasticsearch, j'ai téléchargé le module Wazuh pour Filebeat .

```
root@ahmed:~# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/alerts/
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/archives/
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/module.yml
```

J'ai modifié le fichier **filebeat.yml** pour ajouter le mot de passe précédemment généré pour elastic

```
GNU nano 6.2
# Wazuh - Filebeat configuration file
output.elasticsearch.hosts: ["127.0.0.1:9200"]
output.elasticsearch.password: V4rrrqfBJo7Y0RhpmdHRk

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: false
```

De même, je copie les certificats dans le fichier /etc/filebeat/certs.

```
root@ahmed:~# cp -r /etc/elasticsearch/certs/ca/ /etc/filebeat/certs/
cp /etc/elasticsearch/certs/elasticsearch.crt /etc/filebeat/certs/filebeat.crt
cp /etc/elasticsearch/certs/elasticsearch.key /etc/filebeat/certs/filebeat.key
```

Pour activer et démarrer le service Filebeat, j'ai exécuté les commandes suivantes :

- ✓ systemctl daemon-reload
- ✓ systemctl enable filebeat
- ✓ systemctl start filebeat

Et enfin pour m'assurer que Filebeat a été installé avec succès, j'ai exécuté la commande

filebeat test output

```
root@ahmed:~# filebeat test output
elasticsearch: https://127.0.0.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.17.13
```

3.4 Installation de Kibana :

Pour installer Kibana, j'ai utilisé la commande suivante :

- ✓ apt-get install kibana=7.17.13

J'ai ensuite copié les certificats Elasticsearch dans le dossier de configuration de Kibana , aussi, j'ai téléchargé le fichier de configuration de Kibana puis j'ai le édité pour y ajouter le mot de passe de elastic .

```
GNU nano 6.2                                     /etc/kibana/kibana.yml *
server.host: 0.0.0.0
server.port: 443
elasticsearch.hosts: https://localhost:9200
elasticsearch.password: V4rrqfBJo7Y0RhpmdHRk
```

Ensuite, j'ai créé le répertoire de données de Kibana et j'ai installé le plugin Wazuh Kibana.

```
root@ahmed:~# mkdir /usr/share/kibana/data
root@ahmed:~# chown -R kibana:kibana /usr/share/kibana
root@ahmed:~# cd /usr/share/kibana
root@ahmed:/usr/share/kibana# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.5.4_7.1.7.13-1.zip
Attempting to transfer from https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.5.4_7.17.13-1.zip
Transferring 36404504 bytes.....
Transfer complete
Retrieving metadata from plugin archive
Extracting plugin archive
Extraction complete
Plugin installation complete
```

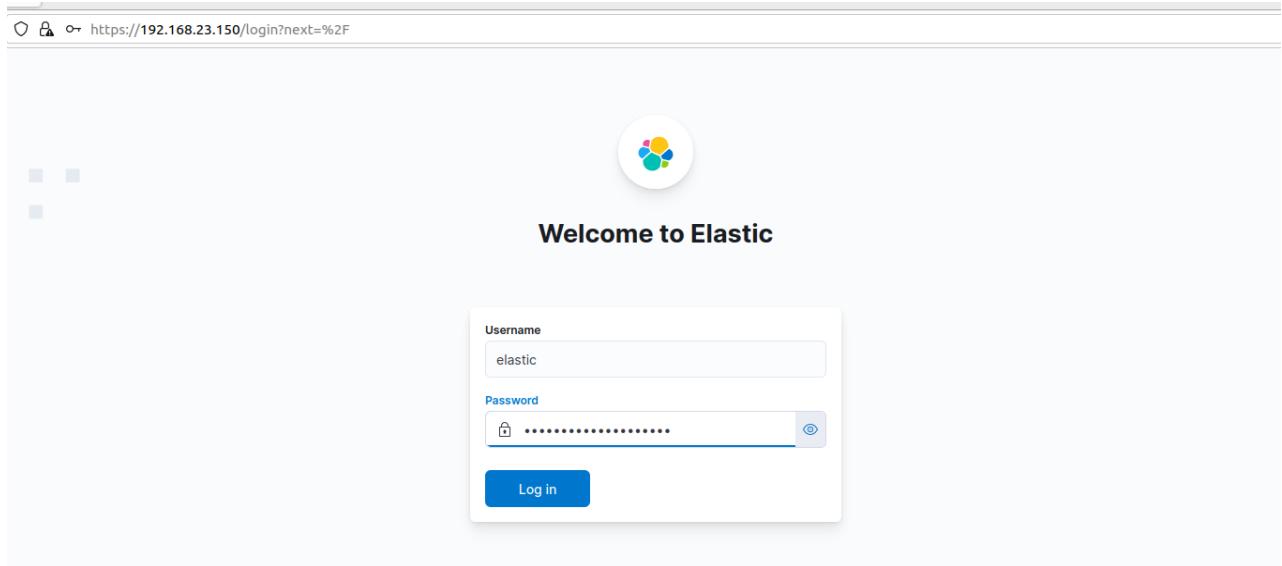
En dernier lieu, j'ai activé et démarré le service Kibana après lier le socket de Kibana au setcap
'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node

```
root@ahmed:/usr/share/kibana# systemctl daemon-reload
root@ahmed:/usr/share/kibana# systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
root@ahmed:/usr/share/kibana# systemctl start kibana
root@ahmed:/usr/share/kibana# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-06-01 08:31:18 CET; 6s ago
     Docs: https://www.elastic.co
 Main PID: 51944 (node)
    Tasks: 11 (limit: 9386)
      Memory: 267.8M
        CPU: 6.730s
       CGroup: /system.slice/kibana.service
               └─51944 /usr/share/kibana/bin/.../node/bin/node /usr/share/kibana/bin/.../src/cli/dist --logging.dest=/var/log/kibana/kibana.log --pid.file=>
08:31:18 01 ↩ ahmed systemd[1]: Started Kibana.
08:31:19 01 ↩ ahmed kibana[51944]: Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable
[Lines 1-13/13 (END)]
```

Pour accéder à l'interface Web nous devons savoir l'adresse IP du serveur wazuh.

```
ahmed@ahmed:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.23.150  netmask 255.255.255.0  broadcast 192.168.23.255
      inet6 fe80::82a4:5628:cea9:1ad1  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:88:bc:8e  txqueuelen 1000  (Ethernet)
          RX packets 644868  bytes 942164731 (942.1 MB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 206520  bytes 12450496 (12.4 MB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Voici une capture qui montre l'authentification au logiciel Elastic Stack, illustrant le processus d'accès sécurisé à l'interface web de Kibana.

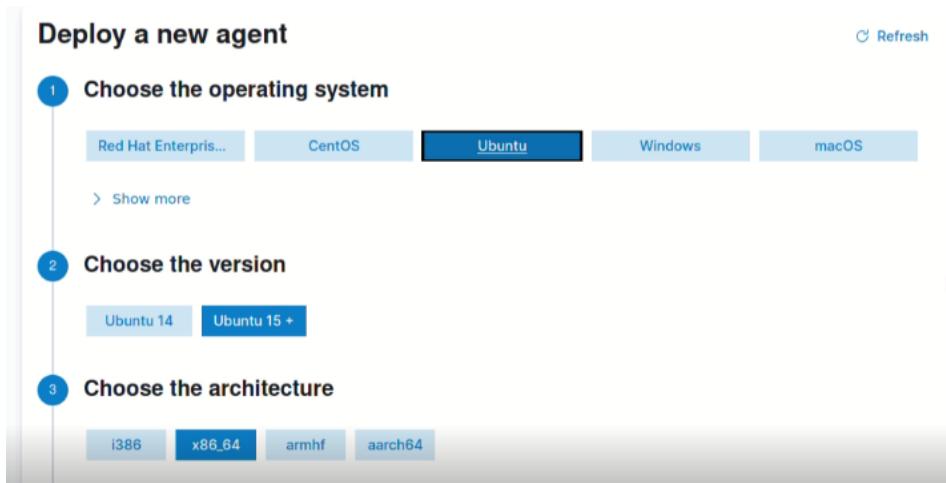


4. Installation des agents :

L'agent Wazuh est multiplateforme et s'exécute sur les terminaux que nous souhaitons surveiller, il communique avec le serveur en envoyant des données en temps quasi réel à travers un canal crypté et authentifié.

4.1 Installation un agent sur Ubuntu :

Pour installer l'agent wazuh, nous allons dans Wazuh > Agents , et choisir déployer un nouvel agent , Ensuite, nous choisissons Ubuntu comme système d'exploitation avec la version et l'architecture correspondantes



Je peux maintenant installer convenablement les agents du wazuh à travers les commandes de la figure ci dessous .

```
root@agent1:~# curl -so wazuh-agent.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.5-1_amd64.deb && sudo WAZUH_MANAGER='192.168.23.150' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent.deb
WAZUH_AGENT_
```

Puis, j'ai activé et démarré l'agent wazuh a l'aide de ces commandes :

- ✓ systemctl daemon-reload
- ✓ systemctl enable wazuh-agent
- ✓ systemctl start wazuh-agent

Il faut tout d'abord verifier son état dans notre machine ubuntu .

```
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-06-02 11:05:24 CET; 7s ago
     Process: 4065 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
       Tasks: 29 (limit: 7041)
      Memory: 73.8M
        CPU: 4.024s
       CGroup: /system.slice/wazuh-agent.service
               ├─4088 /var/ossec/bin/wazuh-execd
               ├─4097 /var/ossec/bin/wazuh-agentd
               ├─4143 /var/ossec/bin/wazuh-syscheckd
               ├─4187 /var/ossec/bin/wazuh-logcollector
               └─4230 /var/ossec/bin/wazuh-modulesd
11:05:18 02 جول agent1 systemd[1]: Starting Wazuh agent...
Lines 1-15
```

Après la vérification de l'état de service agent , je peux vérifier que l'agent est bien ajouter dans le dashboard de serveur wazuh .

The screenshot shows the Wazuh dashboard interface. At the top, there's a navigation bar with the elastic logo, a search bar, and various icons. Below it, a main header says "WAZUH" with a dropdown arrow. The main content area is divided into three main sections: STATUS, DETAILS, and EVOLUTION.

- STATUS:** On the left, there's a large green circle icon. To its right, a legend indicates:
 - Active (1)
 - Disconnected (0)
 - Pending (0)
 - Never connected (0)A "Filter or search agent" input field is below this.
- DETAILS:** This section displays summary statistics:

Active	1
Disconnected	0
Pending	0
Never connected	0
Agents coverage	100.00%

Below these stats are two rows: "Last registered agent" and "Most active agent", each with a double-dot separator.
- EVOLUTION:** This section shows a bar chart titled "Last 24 hours".

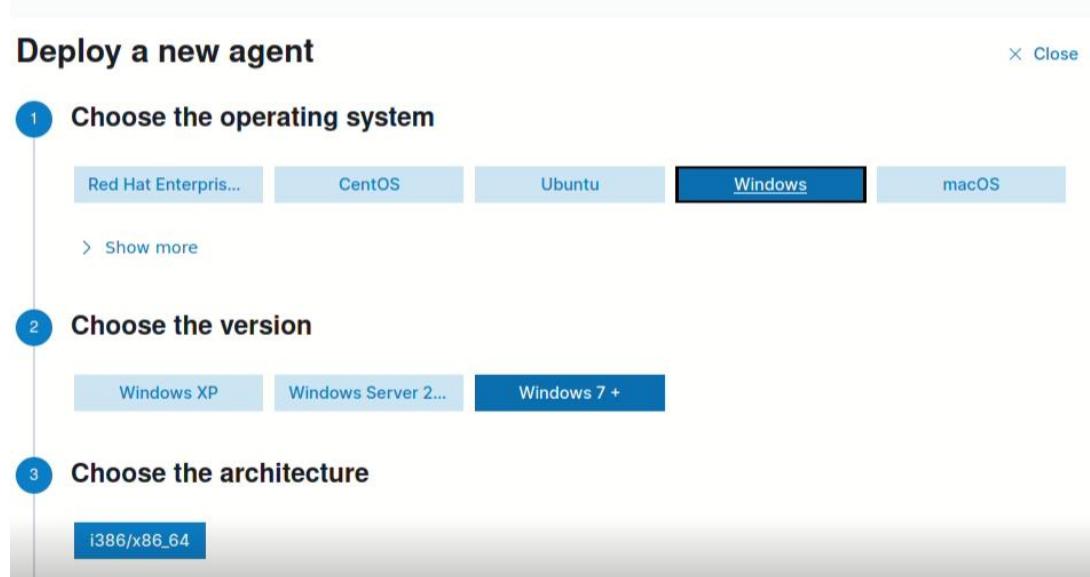
Agents (1) table (highlighted with a red border):

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	agent1	192.168.23.132	default	Ubuntu 22.04.3 LTS	node01	v4.7.5	active	

At the top right of the table, there are buttons for "Deploy new agent", "Export formatted", and "Actions".

4.2 Installation d'un agent sur Windows 10 :

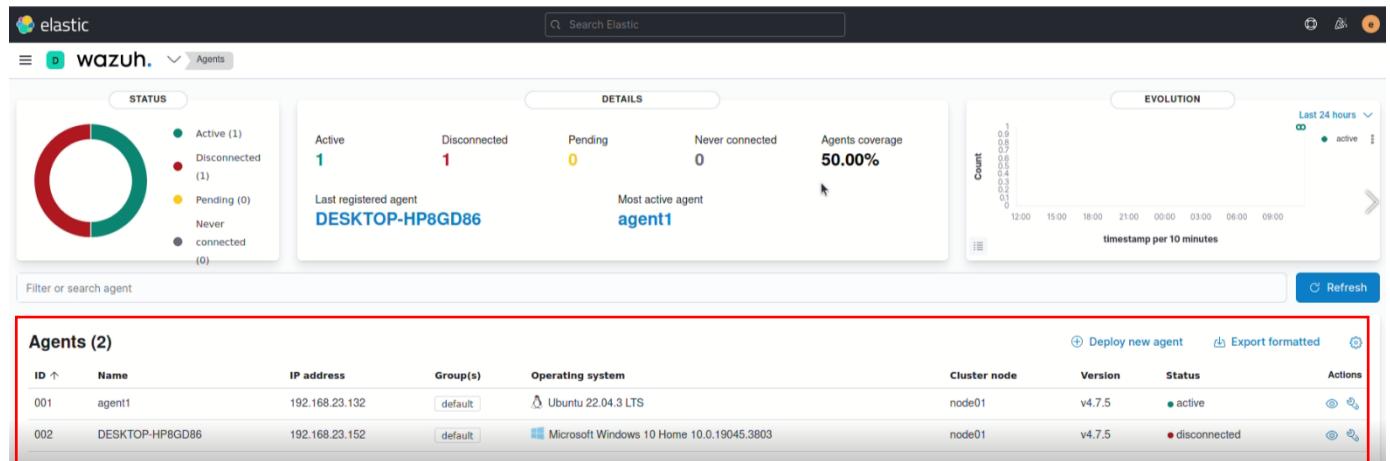
De même , pour déployer un nouvel agent windows ,j'ai remplissé les champs nécessaires comme montre la figure suivante :



Ensuite, à l'aide des deux commandes de la figure ci-dessous, j'ai installé et démarré l'agent wazuh

```
PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.5-1.msi -Outfile $env:tmp\wazuh-agent.msi; msixexec.exe /i $env:tmp\wazuh-agent.msi /q WAZUH_MANAGER='192.168.23.150' WAZUH_REGISTRATION_SERVER='192.168.23.150' WAZUH_AGENT_GROUP='default'
PS C:\Windows\system32> NET Start Wazuh
The Wazuh service is starting.
The Wazuh service was started successfully.
```

Et finalement, j'ai vérifié le succès d'installation à travers les tableaux de bord du wazuh qui affiche l'interconnexion entre le serveur et les clients Ubuntu et Windows et l'activation des agents .



5. Les fonctionnalités clés de Wazuh :

Dans l'interface principale de Wazuh on a 4 catégories :

- Security Information management (Gestion des évènements de sécurité)
- Threat detection and response (Détection et réponse aux menaces)
- Auditing and policy monitoring (Audit et surveillance des politiques de sécurité)
- Regulatory Compliance (Conformité réglementaire)

The screenshot shows the Wazuh interface with a dark header bar. Below it, there are four main sections, each with a red border:

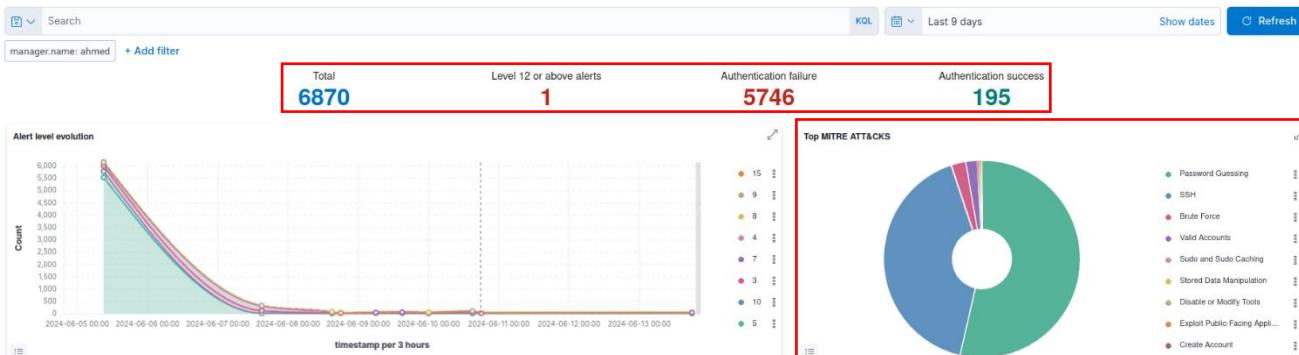
- SECURITY INFORMATION MANAGEMENT**: Contains two items: "Security events" (Browse through your security alerts, identifying issues and threats in your environment) and "Integrity monitoring" (Alerts related to file changes, including permissions, content, ownership and attributes).
- THREAT DETECTION AND RESPONSE**: Contains two items: "Vulnerabilities" (Discover what applications in your environment are affected by well-known vulnerabilities) and "MITRE ATT&CK" (Security events from the knowledge base of adversary tactics and techniques based on real-world observations).
- AUDITING AND POLICY MONITORING**: Contains three items: "Policy monitoring" (Verify that your systems are configured according to your security policies baseline), "System auditing" (Audit users behavior, monitoring command execution and alerting on access to critical files), and "Security configuration assessment" (Scan your assets as part of a configuration assessment audit).
- REGULATORY COMPLIANCE**: Contains three items: "PCI DSS" (Global security standard for entities that process, store or transmit payment cardholder data), "NIST 800-53" (National Institute of Standards and Technology Special Publication 800-53 (NIST 800-53) sets guidelines for federal information systems), and "GDPR" (General Data Protection Regulation (GDPR) sets guidelines for processing of personal data).

5.1 Security Information management :

Le module **Security Information Management** dans Wazuh simplifie la gestion des informations de sécurité , il est formé de deux outils essentiels « Security Events » et « Integrity Monitoring ».

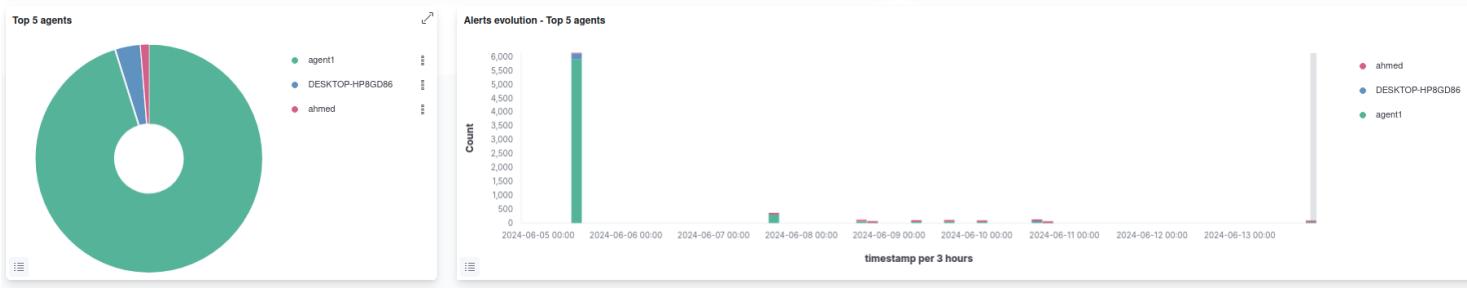
5.1.1 : Security Events :

Le figure ci-dessous répertorie les différents événements de sécurité liés à la machine Ubuntu , Un total de 6870 événements de sécurité ont été détectés sur la machine au cours des dernières 9 jours , incluant 5746 échecs d'authentification et 195 authentifications réussies.



La section "Top MITRE ATT&CK" répertorie les techniques d'attaque les plus couramment observées, basées sur les événements de sécurité détectés par le système pendant cette période .

Le tableau de bord ci-dessous présente la liste des agents générant le plus d'alertes dans ce périodede temps , avec l'agent Ubuntu en tête dans notre cas.



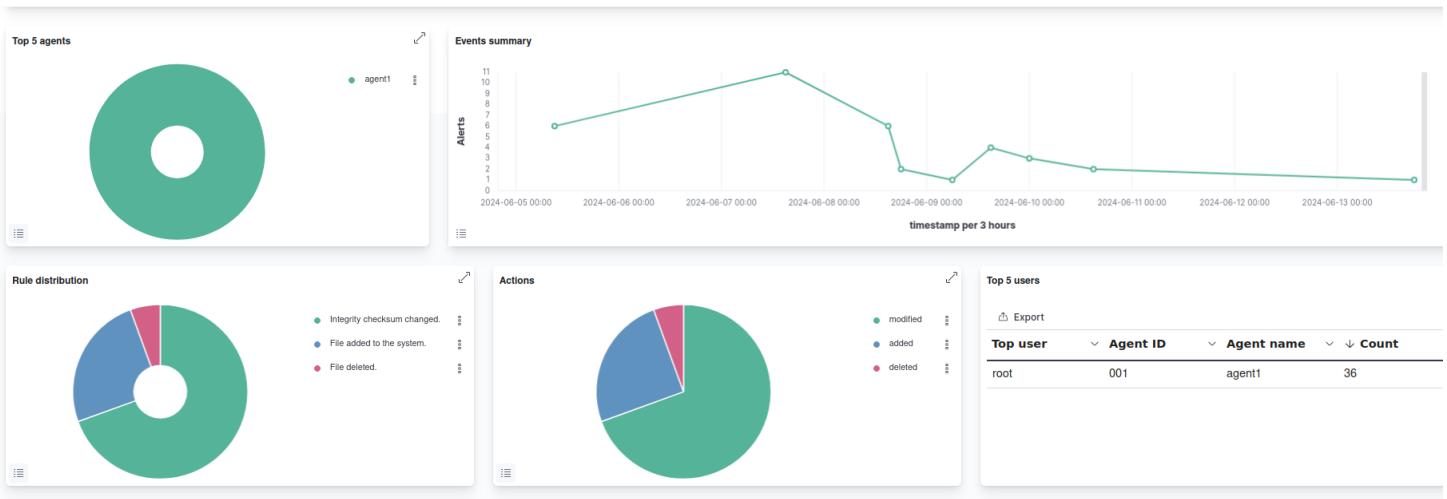
Ensuite, on a la liste des événements que je peux le examiner et le analyser de manière détaillée .

Security Alerts							
Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jun 13, 2024 @ 19:07:48.821	001	agent1	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
> Jun 13, 2024 @ 18:59:26.135	001	agent1			Auditd: SELinux permission check.	3	80730
> Jun 13, 2024 @ 18:59:24.661	001	agent1	T1565.001	Impact	Integrity checksum changed.	7	550
> Jun 13, 2024 @ 18:59:20.149	001	agent1			Auditd: SELinux permission check.	3	80730
> Jun 13, 2024 @ 18:59:20.140	001	agent1			Auditd: SELinux permission check.	3	80730
> Jun 13, 2024 @ 18:59:20.137	001	agent1			Auditd: SELinux permission check.	3	80730
> Jun 13, 2024 @ 18:59:20.134	001	agent1			Auditd: SELinux permission check.	3	80730
> Jun 13, 2024 @ 18:59:20.132	001	agent1			Auditd: SELinux permission check.	3	80730

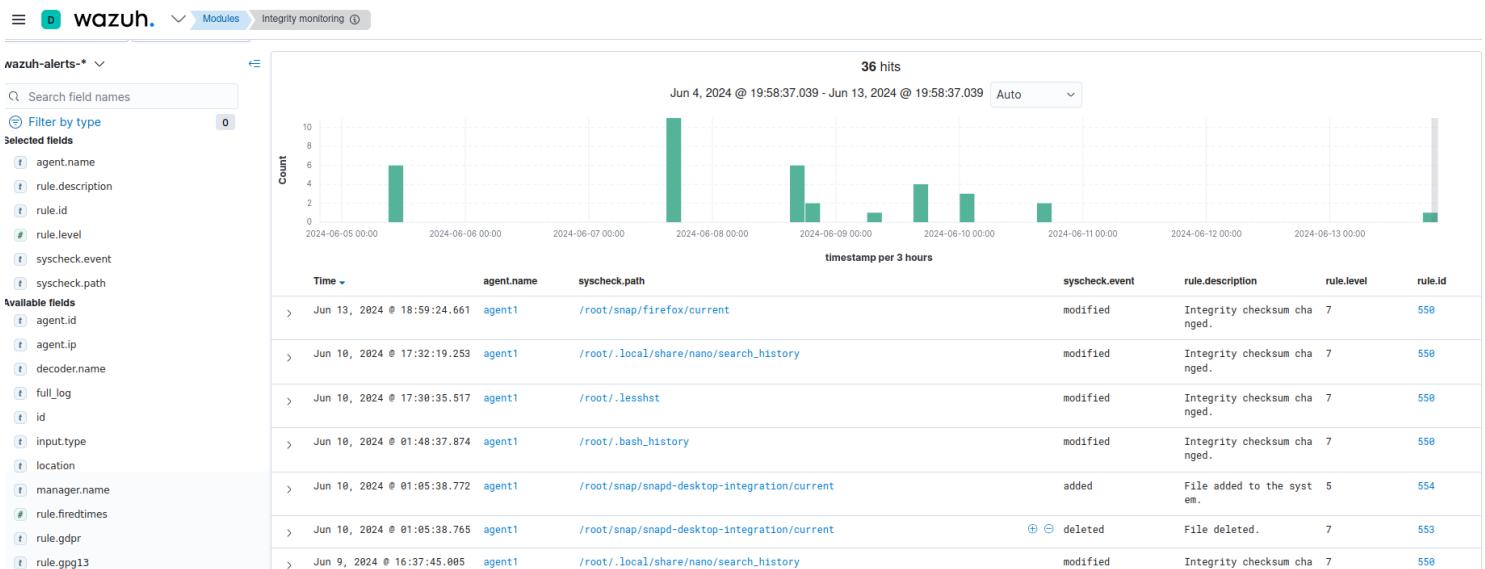
5.1.2 Integrity Monitoring :

Ce module conçu pour surveiller et détecter les modifications non autorisées ou inattendues sur les fichiers critiques, configurations système, registres, et d'autres éléments clés d'un agent.

Le tableau de bord ci-dessous me fournit une première vue graphique sur les principales actions qui ont été détectées sur le client Ubuntu à travers le module du contrôle d'intégrité .



Les informations détaillées concernant ces actions sont accessibles depuis l'onglet ‘ **Events** ’



5.2 Threat detection and response :

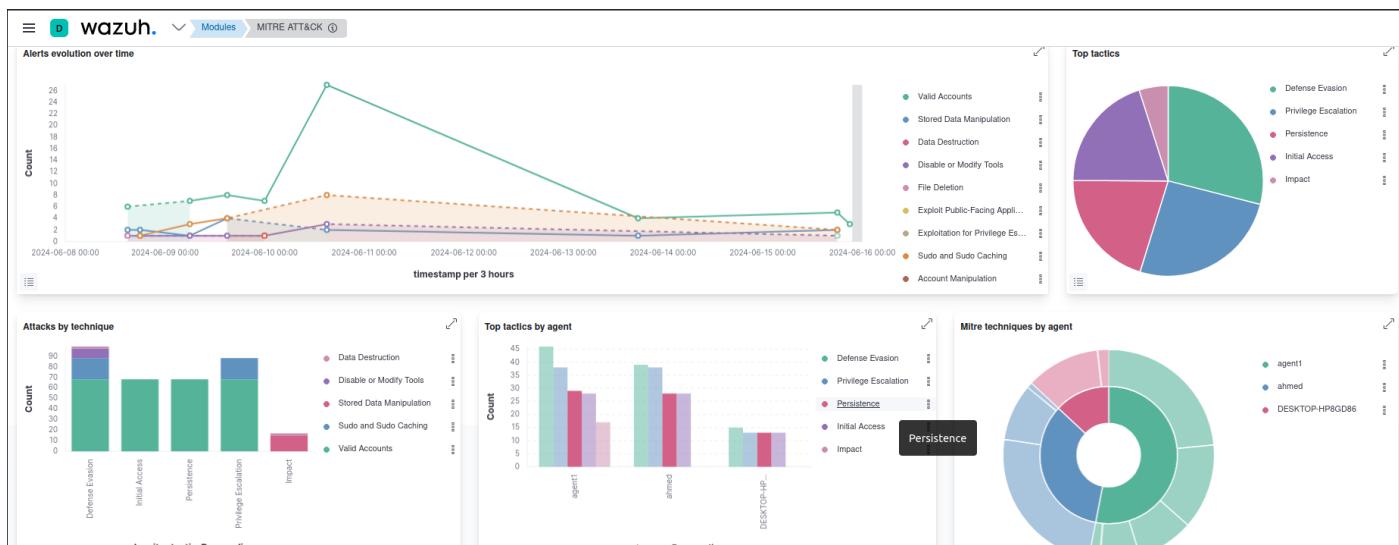
Ce module est composé de deux outils : Vulnérabilités et MITRE ATT&CK

5.2.1 Vulnerabilities :

Le module de détection de vulnérabilités de la plateforme Wazuh a pour objectif d’identifier et de signaler activement les vulnérabilités détectées dans le système concerné.

5.2.2 Mitre & Attack :

Le module MITRE & ATTCK dans Wazuh fournit une puissante capacité d’analyse et de compréhension des attaques en mappant les événements de sécurité aux tactiques et techniques documentées par MITRE & ATTACK.



5.3 Auditing and policy monitoring :

Dans ce module on a trois outils : Policy Monitoring , System Auditing et Security Configuration Assessment .

Le module Auditing and Policy Monitoring de Wazuh permet la surveillance de la conformité des systèmes informatiques aux politiques de sécurité et aux réglementations en vigueur en temps réel.

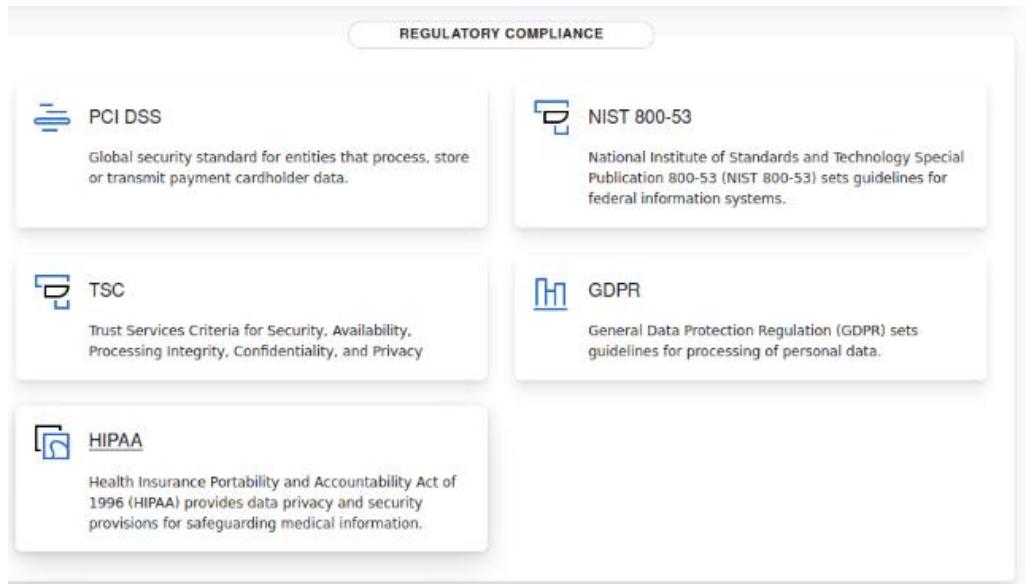
CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0				
Passed	Failed	Not applicable	Score	End scan
125	263	6	32%	Jun 15, 2024 @ 21:37:37.000
Checks (394)				
Filter or search				
ID	Title	Target	Result	
15500	Ensure 'Enforce password history' is set to '24 or more password(s)'.	Command: net.exe accounts	● Not applicable	
15501	Ensure 'Maximum password age' is set to '365 or fewer days, but not 0'.	Command: net.exe accounts	● Not applicable	
15502	Ensure 'Minimum password age' is set to '1 or more day(s)'.	Command: net.exe accounts	● Failed	
15503	Ensure 'Minimum password length' is set to '14 or more character(s)'.	Command: net.exe accounts	● Failed	
15505	Ensure 'Relax minimum password length limits' is set to 'Enabled'.	Registry: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SAM	● Failed	
15506	Ensure 'Account lockout duration' is set to '15 or more minute(s)'.	Command: net.exe accounts	● Failed	
15507	Ensure 'Account lockout threshold' is set to '5 or fewer invalid logon attempt(s), but not 0'.	Command: net.exe accounts	● Failed	
15508	Ensure 'Reset account lockout counter after' is set to '15 or more minute(s)'.	Command: net.exe accounts	● Failed	
15509	Ensure 'Accounts: Administrator account status' is set to 'Disabled'.	Command: net user administrator	● Passed	

Sur le capture ci-dessus, on peut voir un échantillon d'événements indiquant que les meilleures pratiques de sécurisation des mots de passe n'ont pas été mises en place sur l'agent Windows.

5.4 Regulatory Compliance :

Ce module est conçu pour aider les organisations à se conformer aux diverses réglementations et standards de sécurité.

Le jeu de règles par défaut de Wazuh prend en charge divers cadres et normes tels que PCI DSS, HIPAA, NIST 800-53, TSC, et GDPR.



6. Teste de SIEM :

Pour tester le bon fonctionnement du SIEM, j'ai effectué diverses attaques et actions. Ces attaques et actions comprenaient :

- Surveillance de l'intégrité des fichiers
- Détection d'une attaque par force brute SSH
- Détection d'une attaque par injection SQL
- Détection d'une attaque Shellshock
- Détection de processus non autorisés (netcat)

6.1 Surveillance de l'intégrité des fichiers :

J'ai vérifié la capacité du SIEM à détecter les modifications non autorisées des fichiers le module FIM .

Pour configurer ce module j'ai activé dans le fichier de configuration de l'agent , l'audit des données qui détecte toutes informations sur l'utilisateur et le processus modifiés.

```
GNU nano 6.2                               /var/ossec/etc/ossec.conf
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

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

  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>

  <scan_on_start>yes</scan_on_start>

  <!-- Directories to check (perform all possible verifications) -->
```

Le module FIM me affiche toutes les modifications établie sur une fichier , les captures ci-dessous présentent par détails une alerte qui contient tous les données nécessaires.

t _index	wazuh-alerts-4.x-2024.06.16
t agent.id	001
t agent.ip	192.168.23.132
t agent.name	agent1
t decoder.name	syscheck_integrity_changed
t full_log	<div style="border: 1px solid red; padding: 5px;">File '/root/.local/share/nano/search_history' modified Mode: realtime Changed attributes: size,mtime,md5,sha1,sha256 Size changed from '195' to '222' Old modification time was: '1718534242', now it is '1718537259' Old md5sum was: '63d7fc0c9e4d04b1f5eb76426c9a5c43' Now md5sum is: '1d687710a66cc7d17677cc04ba166f9a1'</div>
t id	1718537259.531284
t input.type	log
t location	syscheck
t manager.name	ahmed
t rule.description	Integrity checksum changed.
# rule.firetimes	1
t rule.gdpr	II_5.1.f
t rule.gpg13	4.11

```

t syscheck.event modified
t syscheck.gid_after 0
t syscheck.gname_after root
t syscheck.inode_after 1180774
t syscheck.md5_after 1d687210e5cc7d17677cc94be466f201
t syscheck.md5_before 63d7fc0c9e4d04b1f5eb76426c9a5c43
t syscheck.mode realtime
t syscheck.mtime_after Jun 16, 2024 @ 13:27:39.000
t syscheck.mtime_before Jun 16, 2024 @ 12:37:22.000
t syscheck.path /root/.local/share/nano/search_history
t syscheck.perm_after rw-----
t syscheck.sha1_after c0c03bd8654bf04153710ac341c39be67610ba26
t syscheck.sha1_before 07ad9ed56d5ee8b829e329dd5f13e9d33f0dcf15
t syscheck.sha256_after c9e0b9c2cfca363c6cb5717c4b0994a55734ccdbd7a6cf1f44358ffe3f2c6d45
t syscheck.sha256_before 4931df2828be0d75f02f83715d6d658350773f902e0a642c912b1ce216cf099

```

Cet événement de sécurité indique une modification d'un fichier sur le système , les modifications incluent une augmentation de la taille du fichier, ainsi que des changements dans les sommes de contrôle MD5, SHA1 et SHA256, ce qui indique que le contenu du fichier a été altéré .

6.2 Détection d'une attaque par force brute SSH :

Une attaque par force brute SSH est une méthode où un attaquant tente de se connecter à un serveur SSH en essayant systématiquement toutes les combinaisons possibles de noms d'utilisateur et de mots de passe jusqu'à ce qu'il trouve une combinaison correcte.

Pour simuler cette attaque il est indispensable que le système où l'agent Wazuh est installé dispose également de SSH , j'ai donc activé le service ssh après l'avoir installé dans le système Ubuntu .

```

root@agent1:~# apt install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-server openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 5 newly installed, 0 to remove and 102 not upgraded.
Need to get 1,663 kB of archives.
After this operation, 6,184 kB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

```

root@agent1:~# /lib/systemd/systemd-sysv-install enable ssh
root@agent1:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-06-05 11:22:59 CET; 1min 3s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Main PID: 10315 (sshd)
        Tasks: 1 (limit: 7040)
       Memory: 1.7M
          CPU: 19ms
        CGroup: /system.slice/ssh.service
                  └─10315 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

11:22:59 ٠٥ جولن agent1 systemd[1]: Starting OpenBSD Secure Shell server...
11:22:59 ٠٥ جولن agent1 sshd[10315]: Server listening on 0.0.0.0 port 22.
11:22:59 ٠٥ جولن agent1 sshd[10315]: Server listening on :: port 22.
11:22:59 ٠٥ جولن agent1 systemd[1]: Started OpenBSD Secure Shell server.

```

J'ai effectué ensuite un ping pour vérifier la connectivité entre la machine de hacker (Kali) et la machine de l'agent .

```

└─(ahmed@ahmed)-[~]
└─$ ping 192.168.23.132
PING 192.168.23.132 (192.168.23.132) 56(84) bytes of data.
64 bytes from 192.168.23.132: icmp_seq=1 ttl=64 time=0.602 ms
64 bytes from 192.168.23.132: icmp_seq=2 ttl=64 time=0.289 ms
64 bytes from 192.168.23.132: icmp_seq=3 ttl=64 time=0.335 ms

```

Après avoir vérifié la connectivité entre la machine Kali et la machine de l'agent, j'ai lancé une attaque par force brute à l'aide de l'outil Hydra .

```

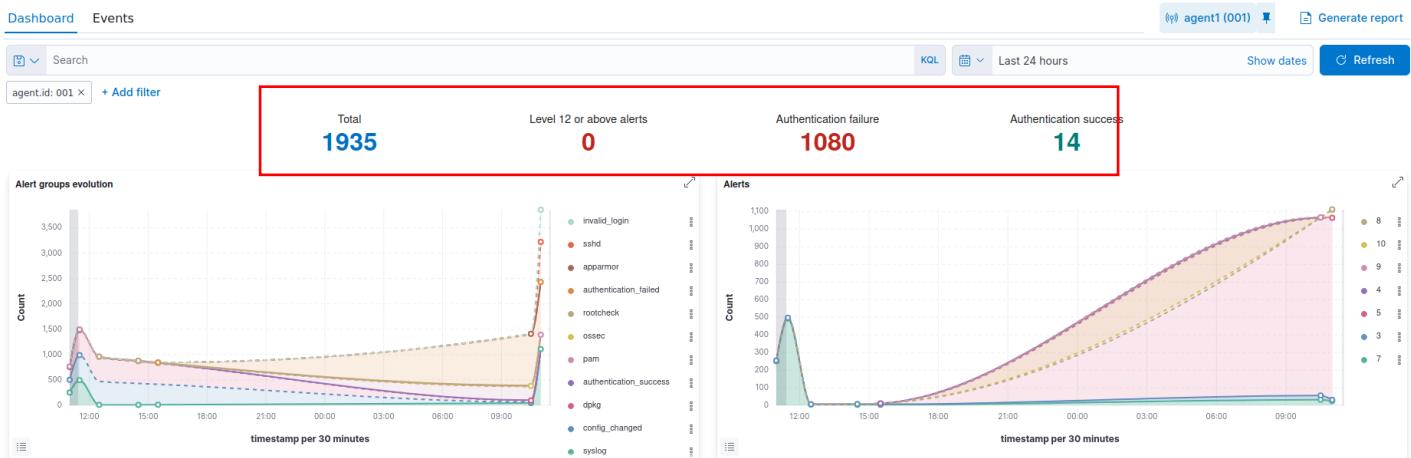
└─(root@ahmed)-[~/hydra]
└─# hydra -L ssh-usernames.txt -P top-20-common-SSH-passwords.txt ssh://192.168.23.132
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret se
rvice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics
anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-16 13:36:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce t
he tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1210 login tries (l:55/p:22), ~76 tries per tas
k
[DATA] attacking ssh://192.168.23.132:22/
[STATUS] 347.00 tries/min, 347 tries in 00:01h, 864 to do in 00:03h, 15 active
└─

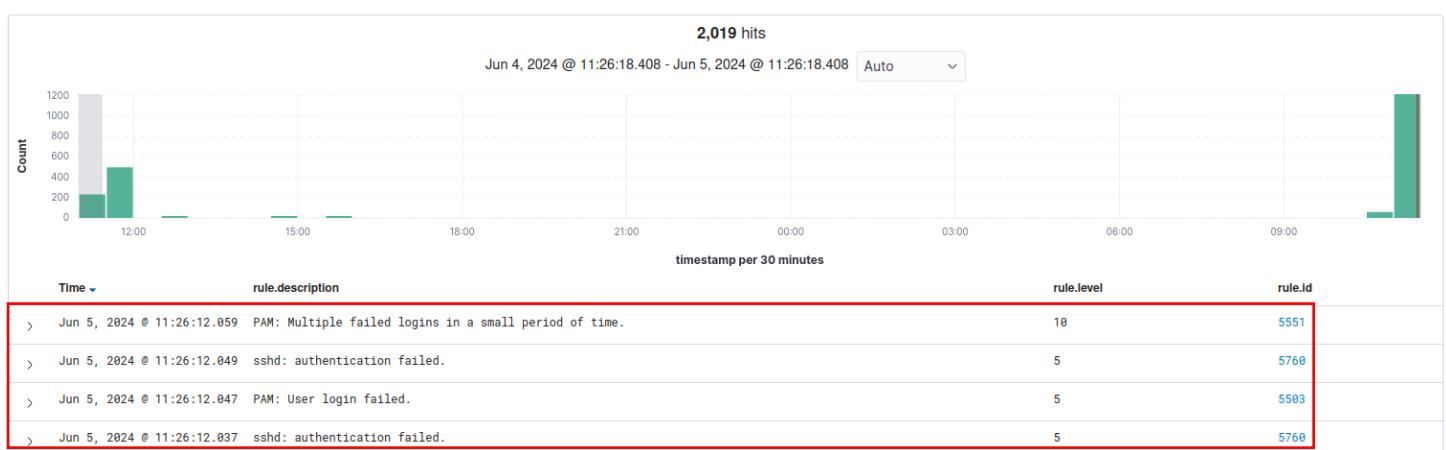
```

J'ai simulé des tentatives répétées de connexion pour accéder à des comptes utilisateur protégés par mot de passe avec l'adresse IP de la cible bien sûr.

L'augmentation nette des événements de sécurité détectés par Wazuh indique une hausse significative des échecs d'authentification ce qui reflète une activité croissante dans les tentatives d'accès non autorisées.



Après avoir confirmé que l'attaque a été enregistrée dans "Security Events", je peux examiner ces événements par détails .



L'événement ci-dessous indique une tentative d'authentification avec le nom d'utilisateur "ahmed" (l'attaquant) provenant de l'adresse IP source 192.168.23.149. Cette tentative ciblait le service SSH exécuté sur l'agent nommé "agent1" avec l'adresse IP 192.168.23.132.

```

t _index           wazuh-alerts-4.x-2024.06.05
t agent.id        001
t agent.ip        192.168.23.132
t agent.name      agent1
t data.dstuser    man
t data.srcip      192.168.23.149
t data.srcport    58620
t decoder.name    sshd
t decoder.parent  sshd
t full_log        Jun 5 11:26:11 agent1 sshd[11371]: Failed password for man from 192.168.23.149 port 58620 ssh2
t id              1717583172.1126988
t input.type      log
t location        /var/log/auth.log
t manager.name    ahmed
t predecoder.hostname  agent1
t predecoder.program_name sshd

```

6.3 Détection d'une attaque par injection SQL :

Une attaque par injection SQL est une méthode où un attaquant insère du code SQL malveillant dans une requête prévue pour une base de données. Cette attaque vise à manipuler ou accéder à des données non autorisées, en exploitant des vulnérabilités dans les applications web qui ne valident pas correctement les entrées des utilisateurs.

Heureusement le module Wazuh offre la possibilité de détecter ce types des attaques nocives

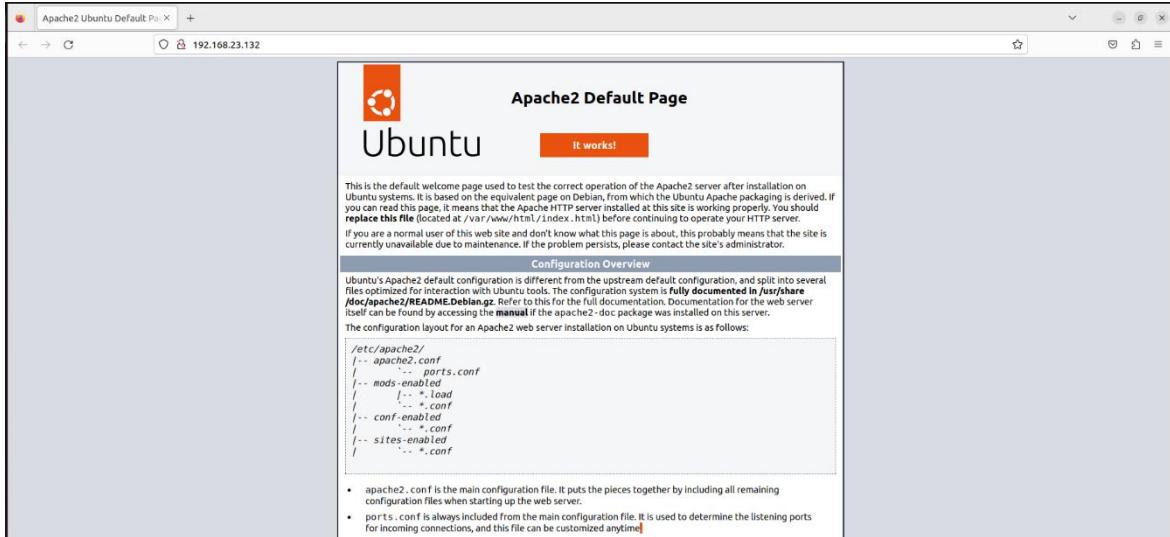
Tout d'abord , j'ai vérifié l'état du service Apache pour confirmer que le serveur Web est en cours d'exécution.

```

root@agent1:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-06-05 10:40:16 CET; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 4767 (apache2)
    Tasks: 55 (limit: 7040)
   Memory: 4.9M
      CPU: 48ms
    CGroup: /system.slice/apache2.service
            └─4767 /usr/sbin/apache2 -k start
                ├─4768 /usr/sbin/apache2 -k start
                ├─4769 /usr/sbin/apache2 -k start

```

La figure ci-dessous affiche l'installation de la page destination d'Apache dans le machine d'agent .



L'agent doit surveiller les logs d'accès de la serveur Apache, pour cela il faut ajouter les lignes suivantes dans le fichier de configuration de l'agent .

```
GNU nano 6.2                               /var/ossec/etc/ossec.conf *

<!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
<logging>
  <log_format>plain</log_format>
</logging>

</ossec_config>
<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

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

  <localfile>
```

Ensuite , et après le redémarrage d'agent j'ai injecté l'attaque SQL Injection comme le montre le figure suivante :

```
(root@ahmed)-[~]
# curl -XGET "http://192.168.23.132/users/?id=SELECT+*+FROM+users";

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.23.132 Port 80</address>
</body></html>
```

Maintenant je peux visualiser les données de l'alerte détectée dans le tableau de bord Wazuh

Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jun 5, 2024 @ 10:55:53.979	T1190	Initial Access	SQL injection attempt.	7	31103

wazuh. Modules agent Security events ⓘ	
Table	JSON
@timestamp	2024-06-05T10:00:22.258Z
_id	PkjX548BELd9XmA8zYqG
agent.id	001
agent.ip	192.168.23.132
agent.name	agent1
data.id	404
data.protocol	GET
data.srcip	192.168.23.149
data.url	/users/?id=SELECT+*+FROM+users
decoder.name	web-accesslog
full_log	192.168.23.149 - - [05/Jun/2024:11:00:20 +0100] "GET /users/?id=SELECT+*+FROM+users HTTP/1.1" 404 437 "-" "curl/8.5.0"
id	1717581622.34204
input.type	log
location	/var/log/apache2/access.log

6.4 Détection d'une attaque Shellshock :

Une attaque Shellshock exploite une vulnérabilité dans le shell Bash de Unix/Linux. L'attaquant insère des commandes malveillantes dans des variables d'environnement, permettant l'exécution non autorisée de code sur le système affecté.

Wazuh est capable de détecter une attaque Shellshock en analysant les journaux du serveur Web collectés .

Tout d'abord j'ai modifié le pare-feu pour permettre l'accès externe aux ports web

```
root@agent1:~# ufw app list
Available applications:
 Apache
 Apache Full
 Apache Secure
 CUPS
 OpenSSH
root@agent1:~# ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
root@agent1:~# ufw status
Status: inactive
```

Puis, j'ai vérifié que le serveur web Apache était en cours d'exécution

```
root@agent1:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Sat 2024-06-08 17:40:51 CET; 1h 29min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 957 (apache2)
      Tasks: 55 (limit: 7040)
     Memory: 7.4M
        CPU: 416ms
      CGroup: /system.slice/apache2.service
              └─957 /usr/sbin/apache2 -k start
                  ├─1223 /usr/sbin/apache2 -k start
                  ├─1224 /usr/sbin/apache2 -k start
```

Pour émuler une attaque, j'ai utilisé la commande ci-dessous depuis la machine kali .

```
[root@ahmed ~]
# sudo curl -H "User-Agent: () { :; }; /bin/cat /etc/passwd" 192.168.23.132
```

Enfin, j'ai visualisé les données d'alerte dans le tableau de bord Wazuh.

Time	rule.description	rule.level	rule.id
> Jun 8, 2024 019:14:59.094	Shellshock attack detected	15	31168

wazuh. Modules agent1 Security events		
t decoder.name t full_log t id t input.type t location t manager.name # rule.firetimes t rule.gdpr t rule.groups t rule.info # rule.mail t rule.mitre.id t rule.mitre.tactic t rule.mitre.technique t rule.nist_800_53 t rule.pc_dss t rule.tsc @ timestamp	t _index wazuh-alerts-4.x-2024.06.08 t agent.id 001 t agent.ip 192.168.23.132 t agent.name agent1 t data.id 200 t data.protocol GET t data.srcip 192.168.23.149 t data.url / t decoder.name web-accesslog t full_log 192.168.23.149 - - [08/Jun/2024:19:14:58 +0100] "GET / HTTP/1.1" 200 10926 "-" "() { ::; }; /bin/cat /etc/passwd" t id 1717870499.41884 # input.type log t location /var/log/apache2/access.log t manager.name ahmed t rule.description Shellshock attack detected # rule.firetimes 1 t rule.gdpr IV_35.7.d	

6.5 Détection de processus non autorisés (netcat) :

Pour détecter les processus non autorisés avec Wazuh, j'ai configuré la surveillance des commandes pour détecter l'exécution de Netcat sur Ubuntu comme une exemple.

Netcat est un utilitaire réseau polyvalent utilisé pour écouter et établir des connexions sur des ports .

Tout d'abord , J'ai ajouté le bloc de configuration suivant dans le fichier de configuration de l'agent Wazuh

```

GNU nano 6.2                               /var/ossec/etc/ossec.conf

</syscheck>

<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <alias>process list</alias>
  <command>ps -e -o pid,uname,command</command>
  <frequency>30</frequency>
</localfile>
<localfile>
  <log_format>audit</log_format>
  <location>/var/log/audit/audit.log</location>
</localfile>

```

J'ai redémarré l'agent Wazuh pour appliquer les modifications et j'ai ensuite installé Netcat avec ces 2 commandes

```

root@agent1:~# sudo systemctl restart wazuh-agent
root@agent1:~# sudo apt install ncat nmap -y

```

Dans le serveur Wazuh J'ai ajouté les règles suivantes dans le fichier **/var/ossec/etc/rules/local_rules.xml** pour déclencher une alerte chaque fois que le programme Netcat se lance .

```

<rule id="100051" level="7" ignore="900">
  <if_sid>100050</if_sid>
  <match>nc -l</match>
  <description>netcat listening for incoming connections.</description>
  <group>process_monitor,</group>
</rule>

```

Sur l'agent Ubuntu , j'ai exécuté la commande suivante pour lancer Netcat en écoute sur le port 8000

```
root@agent1:~# nc -l 8000
```

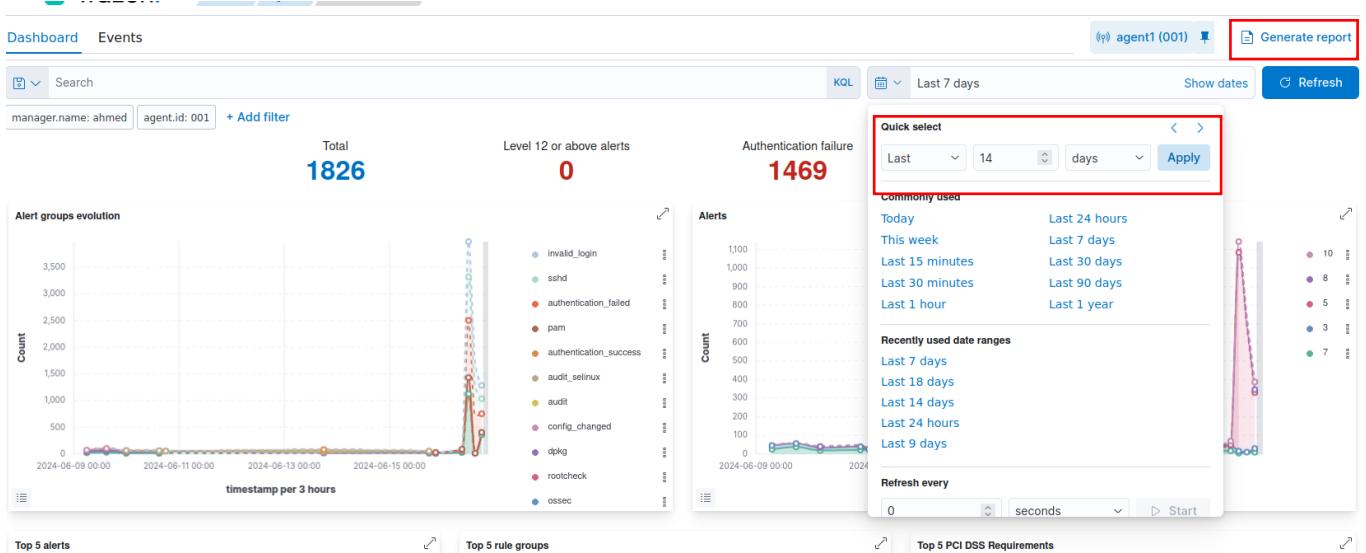
Enfin , J'ai visualisé les données d'alerte dans le tableau de bord Wazuh a travers une événement

```
t _index           wazuh-alerts-4.x-2024.06.09
t agent.id        001
t agent.ip        192.168.23.132
t agent.name      agent1
t decoder.name    ossec
t full_log        >
                  ossec: output: 'process list':
                  PID USER      COMMAND
                  1 root      /sbin/init auto noprompt splash
                  2 root      [kthreadd]
                  3 root      [rcu_gp]
                  4 root      [rcu_par_gp]
                  5 root      [elub flushwa]
t id              1717946213.66475
t input.type      log
t location        process list
t manager.name    ahmed
t rule.description netcat listening for incoming connections.
# rule.firedtimes 1
t rule.groups     local, syslog, sshd, process_monitor
```

7. Génération des rapports :

Grace a Wazuh , je peux créer des rapports détaillés sur les événements de sécurité et la conformité, personnalisables selon les besoins spécifiques, incluant les types d'événements, les périodes et les agents ciblés.

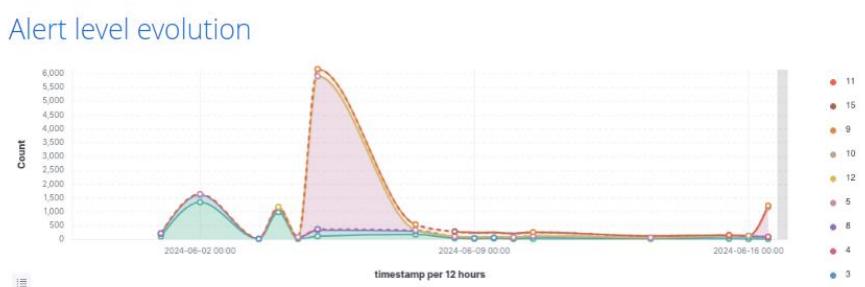
Pour générer ce rapport, j'ai d'abord sélectionné une période spécifique. Ensuite, j'ai cliqué sur 'Generate report', comme illustré dans la figure ci-dessous. Dans mon cas, j'ai choisi une période des 14 derniers jours pour analyser les événements de sécurité et les incidents survenus durant cette période.



Le rapport généré se trouve dans le champ management et peut être facilement téléchargé.

The screenshot shows the Wazuh Management interface under the 'Reporting' tab. It displays a list of generated reports. One report, 'wazuh-module-overview-general-1718556072.pdf', is listed with a size of 161.06KB and a creation date of Jun 16, 2024 @ 17:41:14.119. The 'Actions' column for this report includes a download icon and a delete icon. There is also a 'Refresh' button at the top right. A search bar and a 'Rows per page: 10' dropdown are visible at the bottom.

Ce rapport contient toutes les informations sur les événements de sécurité survenus durant les 14 derniers jours.

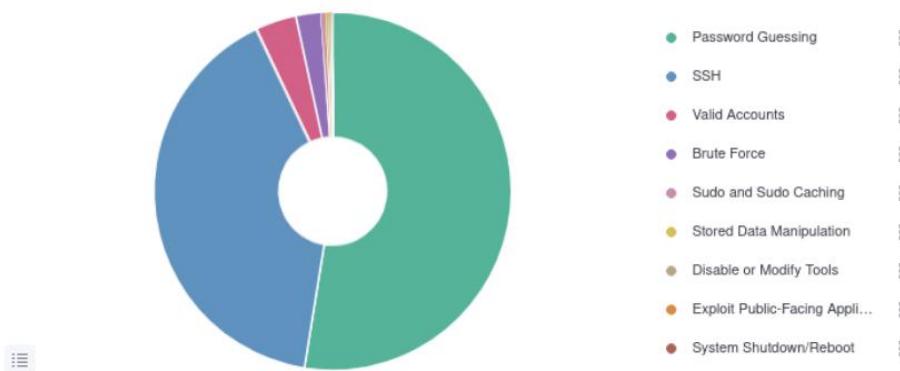


Alerts evolution Top 5 agents



Le rapport offre aussi une vue détaillée de l'évolution des alertes pour les cinq agents les plus actifs au cours de la période sélectionnée. Ce graphique permet d'identifier les tendances et les variations dans le nombre d'alertes générées par chaque agent.

Alerts



Ce dashboard ci-dessus met en lumière les types d'incidents de sécurité les plus récurrents ou les plus graves, permettant de mieux comprendre les principales menaces auxquelles le système est confronté. En analysant ces alertes, on peut identifier les vulnérabilités prioritaires à corriger, ajuster les politiques de sécurité et prendre des mesures préventives pour renforcer la protection de l'infrastructure.

Conclusion Générale du Projet

Ce projet a permis la mise en place d'une solution complète de **SIEM** à travers l'intégration de **Wazuh** avec la **stack ELK** (Elasticsearch, Logstash, Kibana).

Tout au long du développement, nous avons pu :

- Déployer les agents Wazuh pour collecter les événements de sécurité sur les hôtes
- Configurer l'analyse et la corrélation des logs via le moteur de règles Wazuh
- Visualiser les données et les alertes de manière intuitive grâce à **Kibana**
- Automatiser la surveillance de l'infrastructure à travers des tableaux de bord dynamiques

L'intérêt principal d'une solution SIEM comme celle-ci réside dans sa capacité à :

- **Surveiller en temps réel** les activités sur le réseau
- **Déetecter les menaces et comportements suspects** de manière proactive
- **Centraliser et structurer les logs** de différentes sources (serveurs, endpoints, pare-feux...)
- **Alerter immédiatement** en cas d'incident ou d'anomalie
- **Aider à l'analyse post-incident** et à la prise de décision

En conclusion, cette solution améliore considérablement la posture de sécurité de toute organisation.

Elle offre une **visibilité globale**, une **réponse rapide aux incidents**, et une meilleure **maîtrise des risques**