# Mise en œuvre d’un SIEM/SOC pour une API Médicale - Alternance

**Samia Errabite - SOC analyst L1**

---

## Contexte du projet

Dans le cadre du développement d'une API médicale (Fast API/Python), la sécurité des données de santé est une priorité absolue. Une simple faille de code peut permettre à un utilisateur malveillant d'accéder aux dossiers d'autres patients.

L’objectif de ce projet est de mettre en place une architecture de SIEM pour ne plus seulement "subir" les attaques, mais les détecter instantanément via un centre de contrôle (SOC).

---

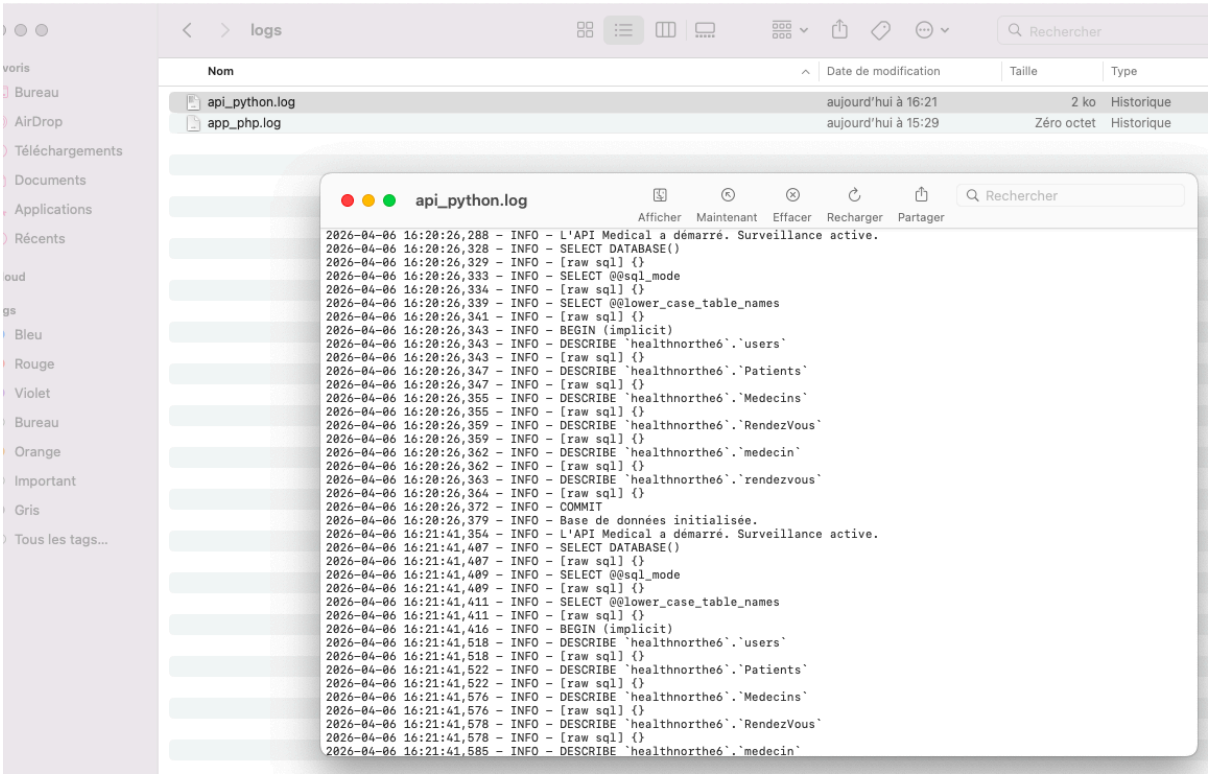
## Mise en place de l'infrastructure

Tout projet de cybersécurité commence par une structure rigoureuse. J'ai commencé par créer une arborescence dédiée pour isoler les flux de données.

- Commandes utilisées : `mkdir` pour les dossiers et `touch` pour initialiser les fichiers de logs.
- Fichiers cibles : `api_python.log` pour mon backend et `app_php.log` pour tester d'autres sources.

Mon API Python a été configurée pour écrire chaque action critique (connexion base de données, démarrage, erreurs) dans le fichier local. On peut voir ici que le système trace chaque opération SQL.

```
[Air-de-Samia:documents samiaerrabite$ cd ..
[Air-de-Samia:~ samiaerrabite$ mkdir -p ~/Documents/Projet_SOC/logs
 Air-de-Samia:~ samiaerrabite$ touch ~/Documents/Projet_SOC/logs/api_python.log
[Air-de-Samia:~ samiaerrabite$ touch ~/Documents/Projet_SOC/logs/app_php.log
 Air-de-Samia:~ samiaerrabite$
```




Pour envoyer ces logs vers le Cloud, j'ai choisi Filebeat. Contrairement à un envoi direct depuis Python, File Beat offre des avantages majeurs :

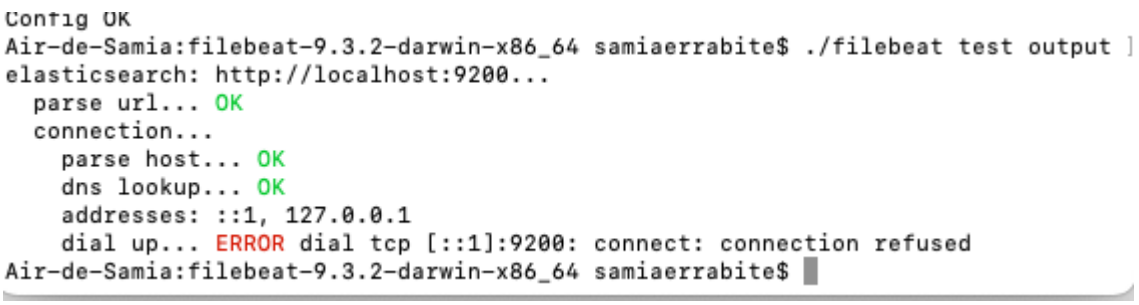
- Agent ultra-léger qui ne ralentit pas l'application.
- En cas de coupure réseau, les logs sont gardés en mémoire

```
# ============================== Filebeat module

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false

  # Period on which files under path should be c
  #reload.period: 10s
```

```
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: projet-soc-samia

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based pat
  paths:
    - /Users/samiaerrabite/Documents/Projet_SOC/logs/*.log
    #- c:\programdata\elasticsearch\logs\*
```

```
Air-de-Samia:Projet_SOC samiaerrabite$ ./filebeat test config
-bash: ./filebeat: No such file or directory
Air-de-Samia:Projet_SOC samiaerrabite$ ls
filebeat-9.3.2-darwin-x86_64    logs
Air-de-Samia:Projet_SOC samiaerrabite$ cd filebeat-9.3.2-darwin-x86_64/
Air-de-Samia:filebeat-9.3.2-darwin-x86_64 samiaerrabite$ ls
LICENSE.txt             filebeat                module
NOTICE.txt              filebeat.reference.yml  modules.d
README.md               filebeat.yml
fields.yml              kibana
Air-de-Samia:filebeat-9.3.2-darwin-x86_64 samiaerrabite$ ./filebeat test config
Config OK
Air-de-Samia:filebeat-9.3.2-darwin-x86_64 samiaerrabite$
```

```
# These settings simplify using Filebeat with the Elastic Cloud (https://clou

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth: "elastic:"
```

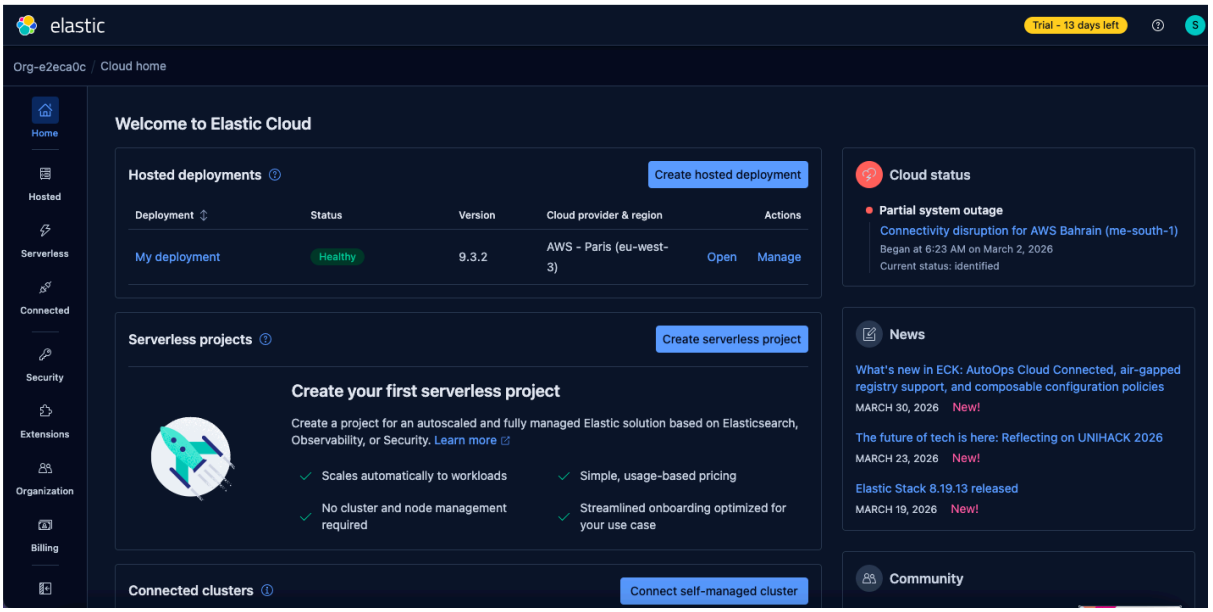




Après avoir récupéré les credentials (`cloud.id` et `cloud.auth`) sur la console Elastic Cloud, je les ai injectés dans le fichier de configuration `filebeat.yml` avant de valider la communication avec le serveur distant via la commande de test, confirmant ainsi que le pont sécurisé entre mon API et mon SOC était opérationnel.

## Centralisation sur Elastic Cloud

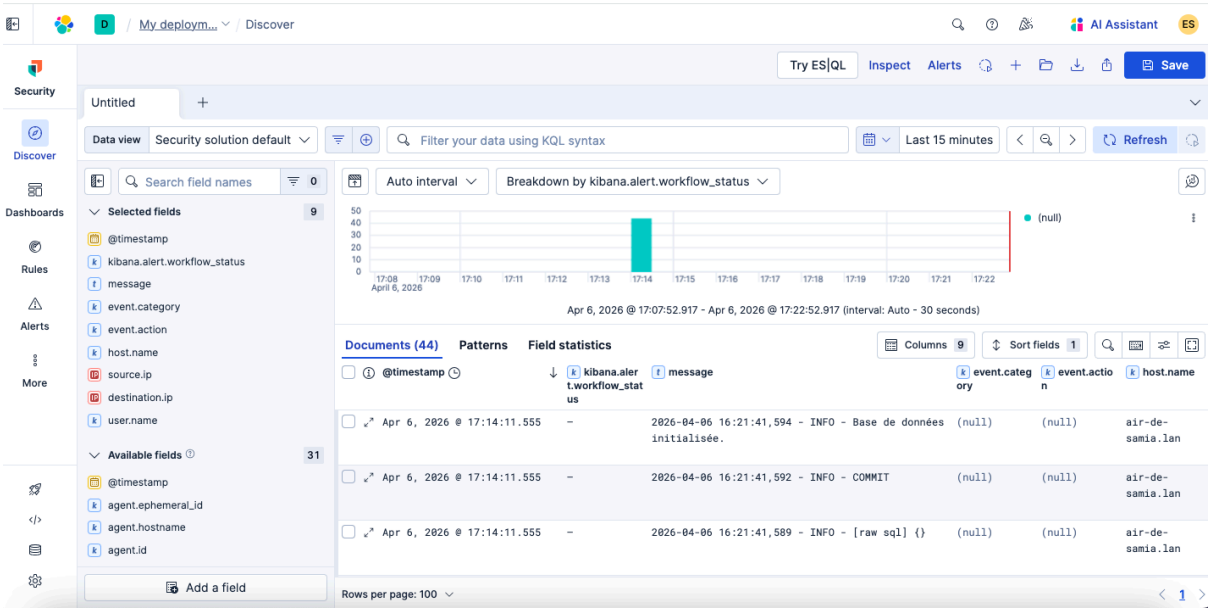
J'ai déployé une instance Elastic Cloud. Cette infrastructure permet de stocker les logs de manière sécurisée et déportée du serveur applicatif.

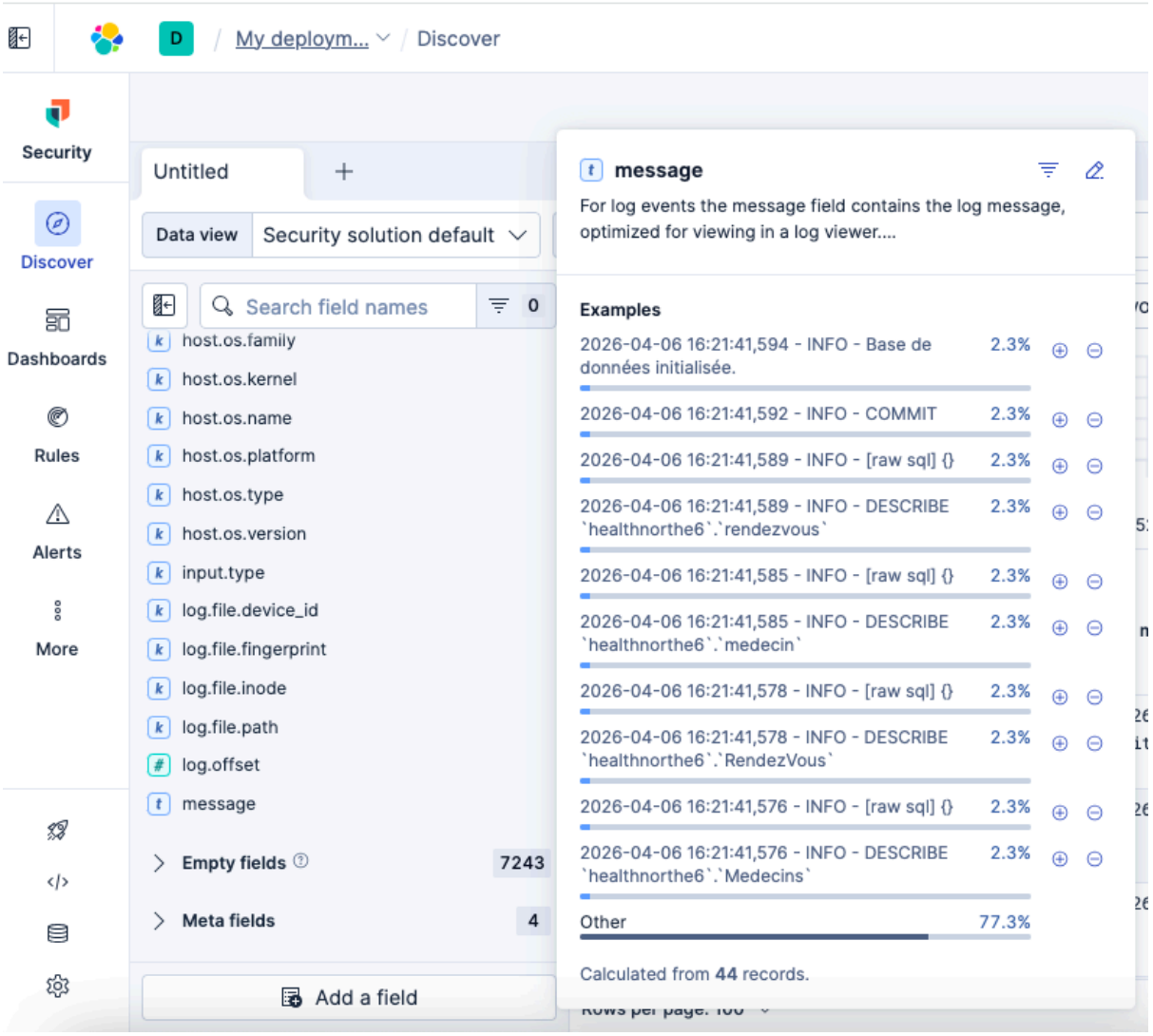


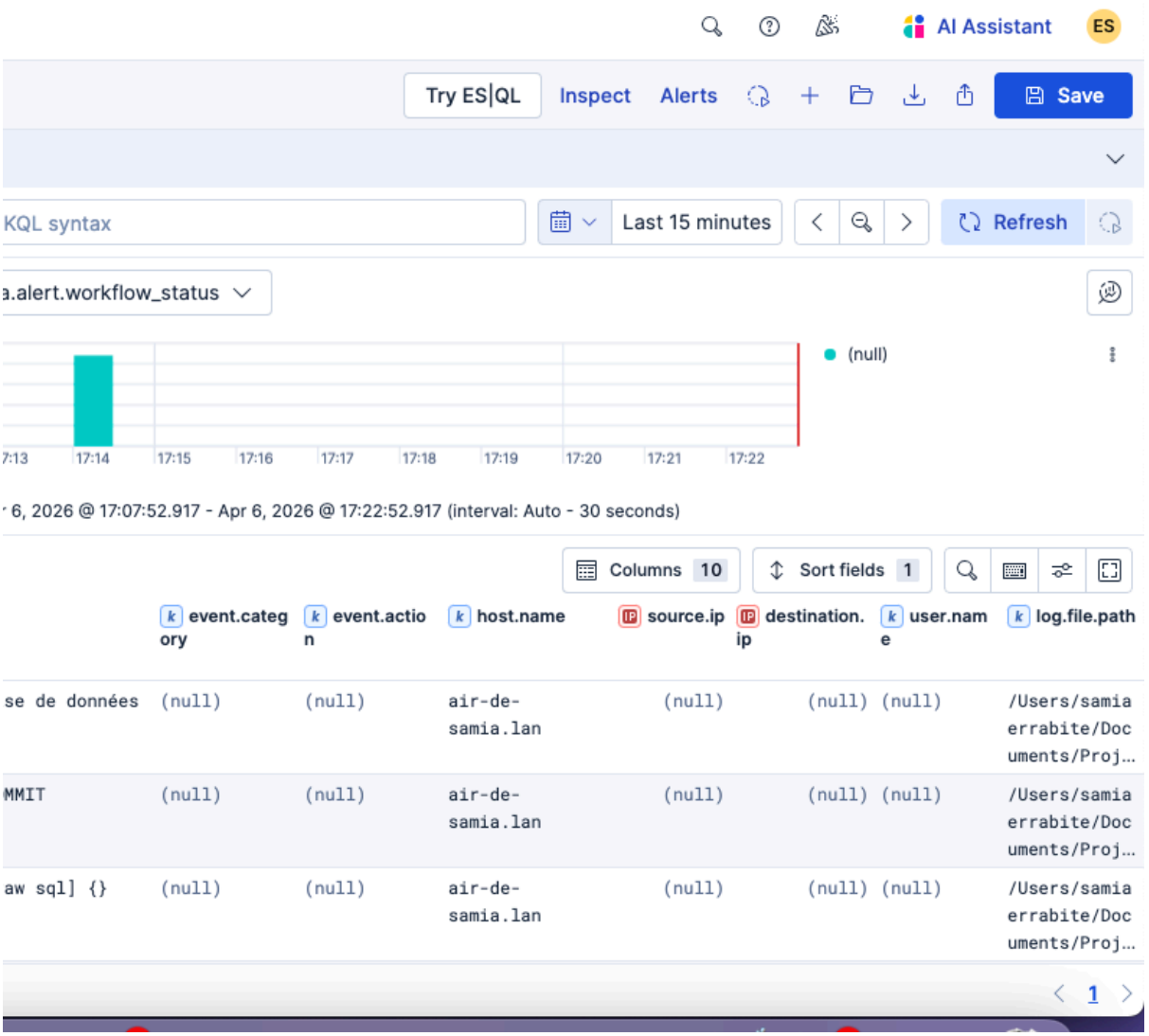

# Détection d'Attaque en Temps Réel (IDOR)

Pour tester l'efficacité de mon SOC, j'ai simulé une attaque IDOR (Insecure Direct Object Reference).

1. Le Code : J'ai ajouté une règle de sécurité qui génère un `logging.warning("ALERTE SOC")` si un utilisateur tente d'accéder à un ID de patient qui n'est pas le sien.
2. La Détection : Dès que l'attaque est lancée, elle remonte instantanément dans Kibana Discover.


My deploym...
Discover
AI Assistant
ES
Security
Try ES|QL
Inspect
Alerts
Save
Untitled
Discover
Data view
Security solution default
Filter your data using KQL syntax
Last 15 minutes
Refresh
Dashboards
Search field names
Auto interval
Breakdown by kibana.alert.workflow_status
Rules
Selected fields
9
@timestamp
kibana.alert.workflow_status
message
event.category
event.action
host.name
source.ip
destination.ip
user.name
Alerts
(null)
Apr 6, 2026 @ 17:07:52.917 - Apr 6, 2026 @ 17:22:52.917 (interval: Auto - 30 seconds)
More
Documents (44)
Patterns
Field statistics
Columns 9
Sort fields 1
@timestamp
kibana.alert.workflow_status
message
event.category
event.action
host.name
Available fields
31
@timestamp
agent.ephemeral_id
agent.hostname
agent.id
Apr 6, 2026 @ 17:14:11.555
2026-04-06 16:21:41,594 - INFO - Base de données initialisée.
(null)
air-de-samia.lan
2026-04-06 16:21:41,592 - INFO - COMMIT
2026-04-06 16:21:41,589 - INFO - [raw sql] {}
Add a field
Rows per page: 100


My deploym...
Discover
Security
Untitled
Discover
Data view
Security solution default
Dashboards
Search field names
0
Rules
host.os.family
host.os.kernel
host.os.name
host.os.platform
host.os.type
host.os.version
Alerts
input.type
log.file.device_id
log.file.fingerprint
log.file.inode
log.file.path
log.offset
More
message
Empty fields
7243
Meta fields
4
Add a field
message
For log events the message field contains the log message, optimized for viewing in a log viewer....
Examples
2026-04-06 16:21:41,594 - INFO - Base de données initialisée. 2.3%
2026-04-06 16:21:41,592 - INFO - COMMIT 2.3%
2026-04-06 16:21:41,589 - INFO - [raw sql] {} 2.3%
2026-04-06 16:21:41,589 - INFO - DESCRIBE `healthnorthe6`.`rendezvous` 2.3%
2026-04-06 16:21:41,585 - INFO - [raw sql] {} 2.3%
2026-04-06 16:21:41,585 - INFO - DESCRIBE `healthnorthe6`.`medecin` 2.3%
2026-04-06 16:21:41,578 - INFO - [raw sql] {} 2.3%
2026-04-06 16:21:41,578 - INFO - DESCRIBE `healthnorthe6`.`RendezVous` 2.3%
2026-04-06 16:21:41,576 - INFO - [raw sql] {} 2.3%
2026-04-06 16:21:41,576 - INFO - DESCRIBE `healthnorthe6`.`Medecins` 2.3%
Other 77.3%
Calculated from 44 records.




## Réponse aux Incidents (Next Steps)

Une fois l'alerte IDOR détectée sur Kibana, la phase de remédiation s'active. Voici les mesures de protection à automatiser :

- Blocage Immédiat : Bannissement automatique de l'adresse IP de l'attaquant au niveau du Pare-feu (WAF).
- Neutralisation du compte : Suspension temporaire du compte utilisateur pour stopper l'exfiltration de données.
- Patching : Correction de la faille dans le code Python (ajout d'une vérification d'autorisation `if user_id != patient_id`).

# Conclusion et déploiement ansible

Ce projet démontre qu'une surveillance active est indispensable pour protéger des données sensibles. Pour une mise en production réelle, ce système est totalement scalable : grâce à Ansible, je pourrais automatiser le déploiement de Filebeat sur des centaines de serveurs en quelques secondes, garantissant une visibilité totale sur toute une infrastructure.

Exemple de playbook :

```
---
- name: Installation et Configuration du SOC Agent (Filebeat)
  hosts: servers_medicals
  become: yes
  vars:
    # On utilise des variables pour ne pas mettre les secrets en clair
    elastic_cloud_id:”."
    elastic_cloud_auth: "elastic:MON_MOT_DE_PASSE_SECURISE"

  tasks:
    - name: 1. Téléchargement de l'agent Filebeat
      apt:
        name: filebeat
        state: present
        update_cache: yes

    - name: 2. Configuration du fichier filebeat.yml
      copy:
        dest: /etc/filebeat/filebeat.yml
        content: |
          filebeat.inputs:
          - type: log
            enabled: true
            paths:
              - /home/user/Documents/Projet_SOC/logs/*.log

          cloud.id: "{{ elastic_cloud_id }}"
          cloud.auth: "{{ elastic_cloud_auth }}"

    - name: 3. Activation et démarrage du service Filebeat
      systemd:
        name: filebeat
        state: restarted
        enabled: yes

    - name: 4. Vérification du statut du SOC
      debug:
        msg: "L'agent Filebeat est déployé et connecté au Cloud Elastic !"
```