Université Mohammed V - Rabat

Master en Cryptographie et Sécurité de l'Information

Projet de Détection d'Intrusion et Réponse à Incident

Implémentation et Analyse avec AIDE sur RHEL 10

Réalisé par :
Younes Lamjaghjagh
Hamza Asimi

Environnement :
Red Hat (ser-secu-aide)
Outil : AIDE v0.18.6

Année Universitaire 2025-2026

Table des matières

1 Introduction et Contexte du Projet **2**
1.1 Objectifs Techniques 2

2 Étape 1 : Préparation de l'environnement **3**
2.1 Partitionnement et Configuration 3
2.2 Création des Utilisateurs 3

3 Étape 2 : Déploiement et Initialisation de AIDE **4**
3.1 Installation du paquet 4
3.2 Initialisation de la base de référence 4

4 Étape 3 : Simulation d'Attaques **5**
4.1 Scénarios d'attaques standard 5
4.2 Scénarios d'attaques avancées (Mécanismes de persistance) 6

5 Étape 4 : Phase de Détection (Alerte AIDE) **7**
5.1 Analyse du Rapport de Détection 7

6 Étape 5 : Investigation et Analyse des Journaux **8**

7 Étape 6 : Remédiation et Restauration **9**
7.1 Éradication des menaces 9
7.2 Restauration des configurations saines 9

8 Étape 7 : Réinitialisation de la Base de Référence **10**

9 Conclusion **11**
9.1 Recommandations de Sécurité 11

1 Introduction et Contexte du Projet

La sécurité des systèmes d'information repose sur des mécanismes de défense en profondeur. Au cœur de ces mécanismes, la vérification de l'intégrité des fichiers systèmes est primordiale pour détecter les compromissions, les portes dérobées (backdoors) et les menaces persistantes avancées (APT).

Ce projet s'inscrit dans le cadre de la mise en place d'un système de détection d'intrusion basé sur l'hôte (HIDS) en utilisant l'outil AIDE (Advanced Intrusion Detection Environment) sur une architecture Red Hat Enterprise Linux 10. L'objectif est d'installer l'environnement, de configurer une base de référence cryptographique (utilisant des algorithmes robustes tels que SHA-256 et SHA-512), de simuler des scénarios d'attaques réalistes, puis de mener une investigation numérique (forensics) sur les journaux du système pour construire une chronologie de l'incident et procéder à la remédiation.

1.1 Objectifs Techniques

- **Préparation** : Installation de RHEL 10 et configuration des comptes de gestion.
- **Initialisation** : Création d'une base de données d'intégrité saine.
- **Simulation** : Exécution d'attaques standard et avancées (persistance, élévation de privilèges).
- **Détection** : Identification des altérations via AIDE.
- **Investigation** : Analyse des logs via `journalctl` et `grep`.
- **Remédiation** : Restauration de l'état sécurisé du système.

2 Étape 1 : Préparation de l'environnement

La première phase du projet a consisté à mettre en place l'environnement de laboratoire. Une machine virtuelle sous Red Hat Enterprise Linux 10 a été déployée.

2.1 Partitionnement et Configuration

Lors de l'installation, un partitionnement manuel LVM a été mis en œuvre pour séparer les répertoires critiques (`/`, `/boot`, `/home`, `/var/log`, et `swap`). Cette ségrégation est une bonne pratique en sécurité pour éviter le déni de service par saturation d'espace disque.

FIGURE 1 – Résumé de l'installation et partitionnement de RHEL 10

2.2 Création des Utilisateurs

Un compte administrateur légitime nommé `admin` a été créé, doté des privilèges d'élévation via le groupe `wheel`. Une politique de mot de passe robuste a été appliquée.

3 Étape 2 : Déploiement et Initialisation de AIDE

Une fois le système de base installé et mis à jour, l'outil AIDE a été déployé. AIDE fonctionne en créant une base de données de référence à partir des règles définies dans son fichier de configuration (`/etc/aide.conf`).

3.1 Installation du paquet

L'installation s'est faite via le gestionnaire de paquets de Red Hat :

```
sudo dnf install aide -y
```

3.2 Initialisation de la base de référence

La génération de la base de données initiale est une étape critique. Elle "photographie" l'état sain du système en calculant les hachages cryptographiques de tous les fichiers surveillés.

```
sudo aide --init
sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

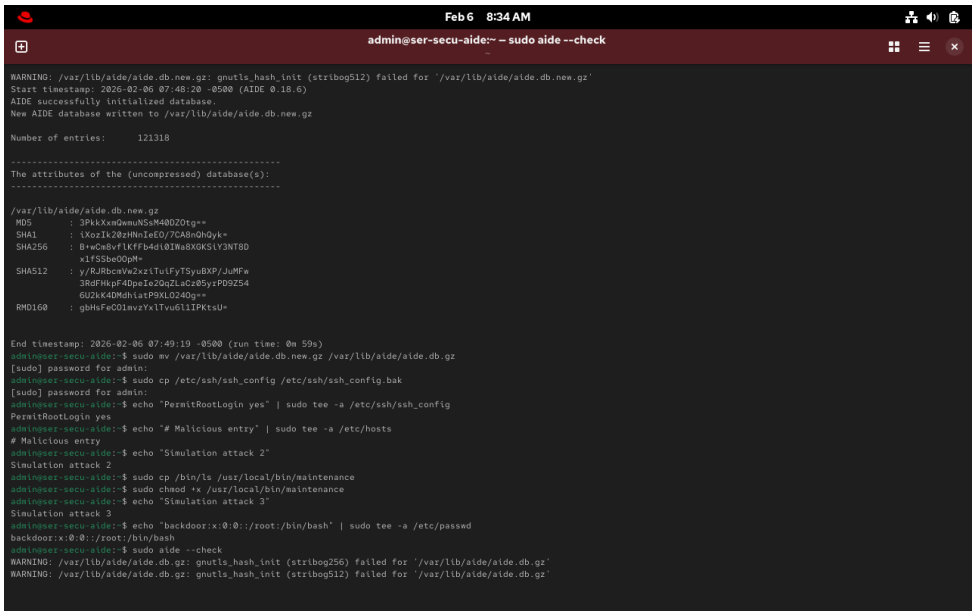

FIGURE 2 – Initialisation réussie de la base de données AIDE (121 318 entrées)

4 Étape 3 : Simulation d'Attaques

Pour éprouver le système de détection, le serveur a fait l'objet de plusieurs modifications non autorisées simulant le comportement d'un attaquant ayant compromis les identifiants d'un administrateur.

4.1 Scénarios d'attaques standard

Ces premières attaques visent à affaiblir les défenses périphériques et à laisser des portes dérobées évidentes.

1. **Altération SSH** : Modification du fichier de configuration SSH pour autoriser une connexion directe avec le compte root (`PermitRootLogin yes`).
2. **Empoisonnement DNS local** : Injection d'une fausse entrée malveillante dans le fichier `/etc/hosts` pour rediriger le trafic.
3. **Déploiement d'un binaire caché** : Copie de l'utilitaire `/bin/ls` vers `/usr/local/bin/mainter` et application des droits d'exécution. L'attaquant tente de camoufler un outil sous un nom anodin.
4. **Création d'un utilisateur fantôme** : Injection directe dans `/etc/passwd` d'un utilisateur nommé `backdoor` possédant l'UID 0 (privilèges root).

Ces actions ont été exécutées via des combinaisons de commandes `echo` et `sudo tee -a` afin de forcer l'écriture dans des répertoires restreints, simulant une élévation de privilèges réussie.

4.2 Scénarios d'attaques avancées (Mécanismes de persistance)

Pour élever le niveau de difficulté de l'investigation, des techniques de type APT (Advanced Persistent Threat) ont été déployées. L'objectif de l'attaquant ici n'est plus seulement de rentrer, mais de rester de manière indétectable, même en cas de redémarrage ou de changement de mots de passe.

- **Persistance via Tâche Planifiée (Cron)** : Modification du fichier `/etc/crontab` pour forcer l'exécution automatique du binaire camouflé (`maintenance`) toutes les minutes.
- **Élévation de privilèges silencieuse (Sudoers)** : Création d'un fichier discret `/etc/sudoers.d/admin_backdoor` accordant les droits `root` absolus et sans mot de passe (`NOPASSWD: ALL`).
- **Faux Service de Démarrage (Systemd)** : Implantation d'un fichier `/etc/systemd/system/sy` Ce service se lance au démarrage de la machine, garantissant la survie de la porte dérobée aux redémarrages.

Ces vecteurs ciblent des composants fondamentaux de Red Hat (Systemd et Cron) que peu d'administrateurs surveillent quotidiennement, démontrant l'intérêt d'un HIDS comme AIDE.

5 Étape 4 : Phase de Détection (Alerte AIDE)

À la suite de la compromission, une analyse d'intégrité a été lancée manuellement pour vérifier si AIDE identifiait les modifications. L'algorithme a recalculé les hachages de l'ensemble du système pour les confronter à sa base saine.

```
sudo aide --check
```

5.1 Analyse du Rapport de Détection

Le rapport généré par AIDE a été catégorique (*AIDE found differences between database and filesystem*). Il a identifié avec succès 4 entrées ajoutées et 11 entrées modifiées.

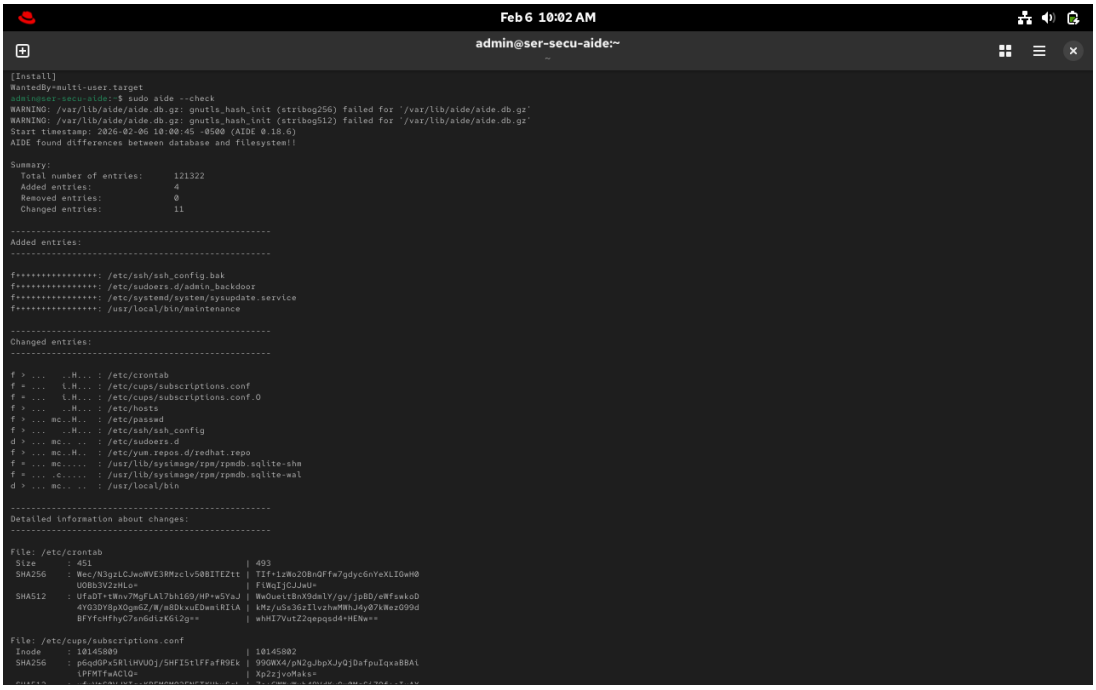

FIGURE 3 – Détection par AIDE : Identification des fichiers de persistance ajoutés

Les signatures SHA-256 et SHA-512 ont divergé, prouvant formellement que l'intégrité des fichiers `/etc/crontab`, `/etc/passwd` ou encore `ssh_config` avait été rompue. AIDE a également détecté les changements d'Inodes et d'horodatages (Mtime/Ctime).

6 Étape 5 : Investigation et Analyse des Journaux

La détection n'est que la première étape de la réponse à incident. Il a fallu ensuite fouiller les journaux système pour identifier *qui* a fait les modifications, *comment*, et *quand*.

Sur RHEL 10, les événements de sécurité sont principalement gérés par `systemd` via `journalctl`, ainsi que dans `/var/log/secure`. L'investigation s'est concentrée sur les commandes `sudo` exécutées.

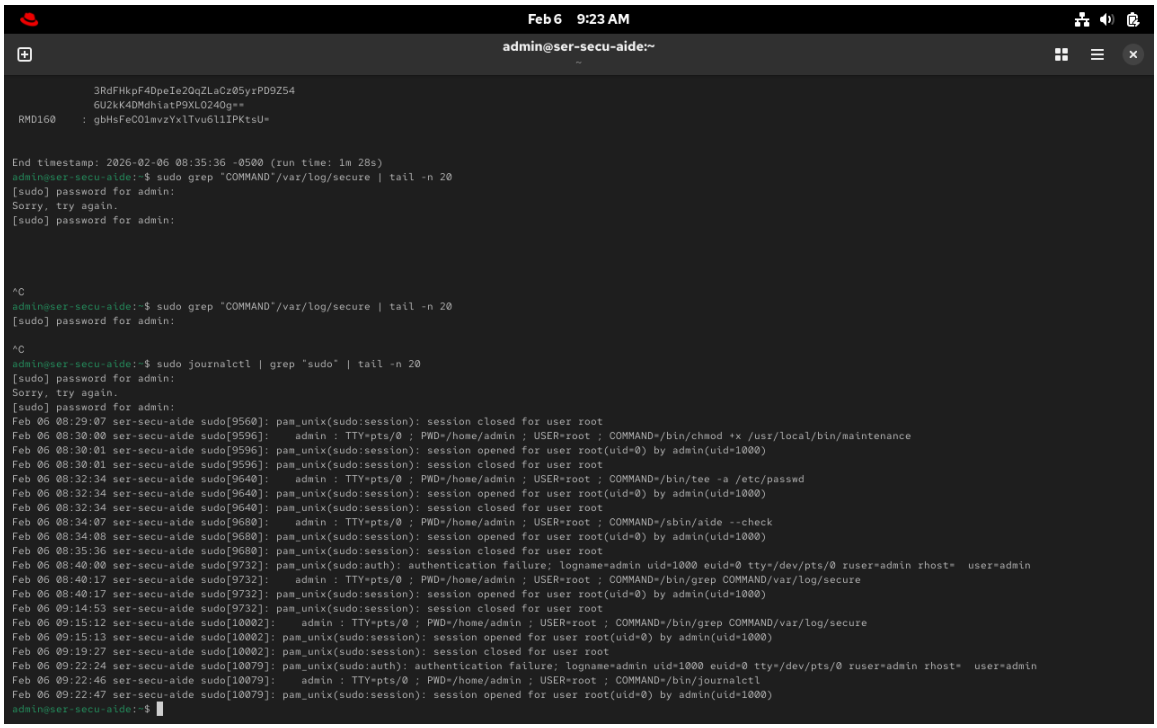

FIGURE 4 – Extraction des traces d'attaques horodatées via journalctl

L'analyse des logs a permis d'extraire les horodatages exacts (timestamps) de l'exécution des commandes `chmod`, `tee -a`, et `cp`, prouvant que le compte de l'utilisateur légitime `admin` avait été utilisé pour exécuter l'attaque.

7 Étape 6 : Remédiation et Restauration

L'incident ayant été identifié et documenté, la phase d'éradication a été enclenchée pour nettoyer le serveur Red Hat.

7.1 Éradication des menaces

Toutes les portes dérobées ont été supprimées du système :

```
sudo rm -f /usr/local/bin/maintenance
sudo rm -f /etc/sudoers.d/admin_backdoor
sudo rm -f /etc/systemd/system/sysupdate.service
```

7.2 Restauration des configurations saines

Les configurations corrompues ont été purgées de leurs lignes malveillantes via l'éditeur de flux `sed`, et le fichier SSH a été restauré depuis sa sauvegarde (`.bak`).

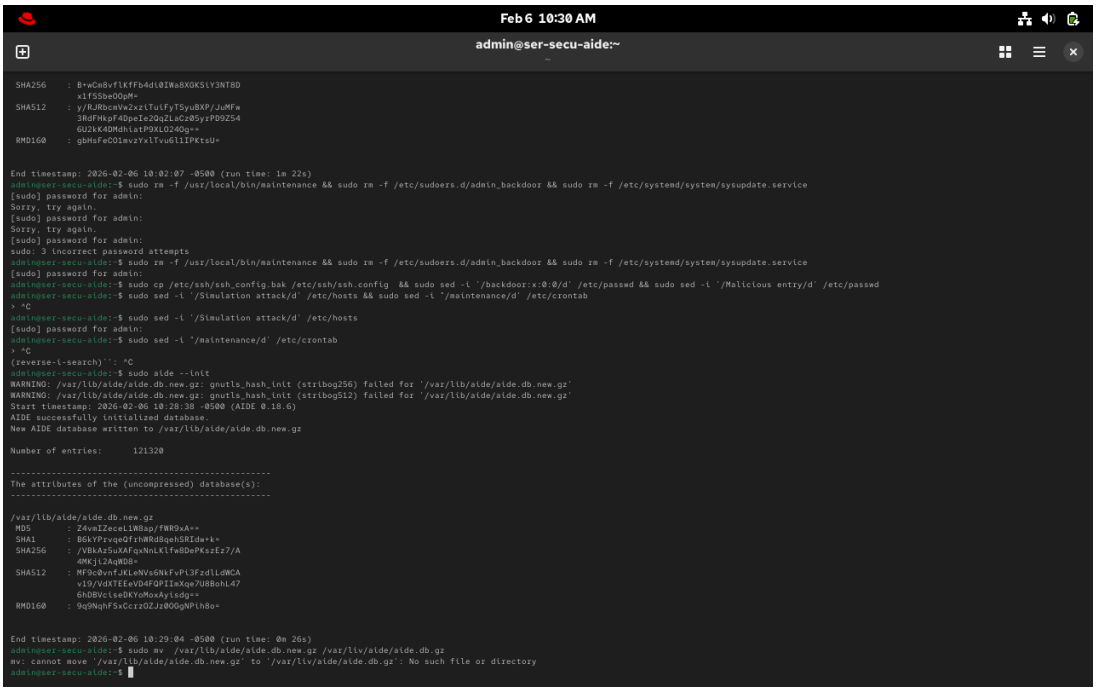

FIGURE 5 – Phase de remédiation : Nettoyage des fichiers et restauration

8 Étape 7 : Réinitialisation de la Base de Référence

Le système étant de nouveau intègre, propre, et libre de tout mécanisme de persistance, il a été nécessaire de clore l'incident.

Pour ce faire, une nouvelle empreinte complète du système a été générée. Si cette étape est omise, l'outil AIDE continuera de signaler les suppressions de backdoors comme des altérations anormales lors du prochain contrôle.

```
sudo aide --init
sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

Cette action a permis de sceller le nouvel état de sécurité du serveur `ser-secu-aide`. Le HIDS est désormais calibré pour détecter toute nouvelle anomalie future basée sur ce système purifié.

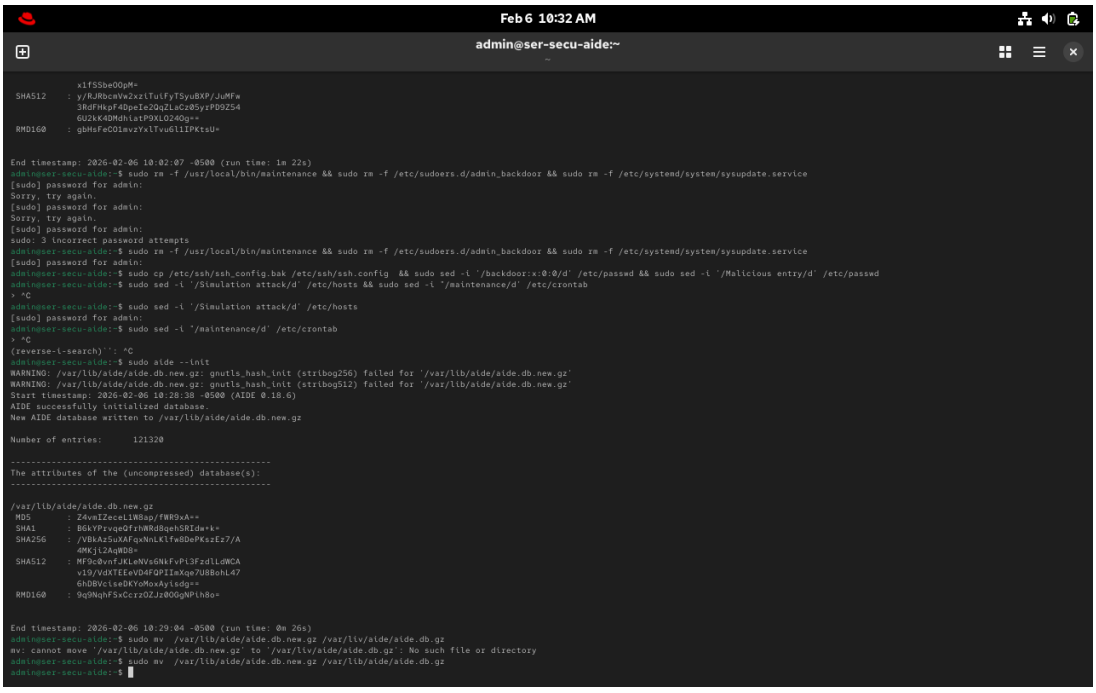

FIGURE 6 – Clôture de l'incident : Génération de la nouvelle base AIDE post-nettoyage

9 Conclusion

Ce projet d'implémentation sur Red Hat Enterprise Linux 10 démontre l'efficacité absolue des systèmes HIDS basés sur le contrôle d'intégrité (FIM - File Integrity Monitoring). L'outil AIDE a su détecter avec une précision cryptographique non seulement les attaques évidentes (fichiers modifiés), mais également les techniques furtives de persistance (APT) dissimulées dans les profondeurs du système de gestion des services (`systemd`).

L'investigation subséquente a mis en exergue l'importance de la journalisation centralisée et robuste. Sans des logs précis conservés par `journald`, il aurait été impossible de déterminer la chronologie de l'attaque.

9.1 Recommandations de Sécurité

À l'issue de cet incident, les mesures suivantes doivent être appliquées pour durcir l'infrastructure :

- **Principe de moindre privilège (Sudoers) :** Restreindre drastiquement les commandes autorisées pour le groupe `wheel`. L'utilisateur n'a pas besoin d'un accès `ALL=(ALL)` systématique.
- **Automatisation de AIDE :** Mettre en place une tâche planifiée légitime (`cron`) qui exécute `aide -check` toutes les 4 heures avec l'envoi d'un rapport par e-mail au SOC (Security Operations Center).
- **Protection des journaux :** Configurer `rsyslog` ou `journald` pour transmettre les événements en temps réel vers un serveur distant. Si l'attaquant avait décidé d'effacer `/var/log/secure`, l'investigation aurait été aveugle.
- **Contrôle SSH :** Implémenter une authentification par clés asymétriques exclusivement et désactiver totalement la connexion par mot de passe.

Fin du rapport d'incident.