UNIVERSITÉ MOHAMMED V DE RABAT

Faculté des Sciences

Département d'Informatique

Filière Cybersécurité intelligente et technologies émergentes

---

Gestion des vulnérabilités et configurations sécurisées

Réalisé par :
Nada Maach

Encadré par :
Pr. Oussama SBAI

Année Universitaire 2024-2025

Table des matières

Objectif

Ce rapport vise à identifier et analyser les vulnérabilités critiques détectées sur deux serveurs de l'entreprise, évaluer leur impact, et proposer des correctifs conformes aux standards de sécurité (CVE/NVD, CIS Benchmarks). Enfin, une méthodologie d'audit sera mise en place pour garantir l'efficacité des mesures appliquées

Rapport de Sécurité : Identification et Analyse des Vulnérabilités

3 avril 2025

0.1 Partie 1 : Identification et analyse des vulnérabilités

0.1.1 CVE-2022-22720

Conditions techniques d'exploitation

La vulnérabilité CVE-2022-22720 a un score CVSS de 9.8, ce qui signifie qu'elle est critique. Elle affecte le serveur Apache HTTPD avant la version 2.4.52 et est due à une mauvaise gestion de la fermeture des connexions entrantes lors du rejet du corps d'une requête HTTP.

Les conditions d'exploitation sont :

— Version vulnérable : Apache HTTPD 2.4.52
— Protocole concerné : Requêtes HTTP avec corps de requête mal géré
— Aucun privilège requis : L'attaque peut être réalisée à distance sans authentification
— Vecteur d'attaque : Un attaquant pourrait envoyer des requêtes spécialement conçues pour provoquer un comportement anormal du serveur
— Aucune interaction utilisateur nécessaire

Impacts potentiels pour l'organisation en cas d'exploitation réussie

Si cette vulnérabilité est exploitée, elle peut entraîner des conséquences graves pour l'organisation. Elle peut provoquer un déni de service (DoS), rendant les serveurs inaccessibles et perturbant les opérations critiques. Une attaque par request smuggling peut permettre aux attaquants d'accéder à des ressources sensibles en contournant les mécanismes de sécurité, entraînant un risque de fuite de données. De plus, un dépassement de tampon (buffer overflow) peut permettre l'exécution de code malveillant, compromettant ainsi l'intégrité et la confidentialité du système.

Ces impacts techniques se traduisent par des pertes financières importantes dues à l'indisponibilité des services, à la mobilisation des équipes pour la remédiation et aux éventuelles sanctions réglementaires. L'image de l'organisation peut également être affectée, réduisant la confiance des clients et partenaires.

0.1.2 Configuration non sécurisée

Conditions techniques d'exploitation

Une configuration non sécurisée sur un serveur Linux Ubuntu expose des services obsolètes et vulnérables, tels que FTP (port 21) et Telnet (port 23). L'exploitation repose sur plusieurs conditions :

— Services actifs : FTP et Telnet doivent être installés et en cours d'exécution.
— Ports ouverts : Les ports 21 et 23 doivent être accessibles depuis l'extérieur.
— Absence de chiffrement : FTP et Telnet transmettent les données en clair, facilitant les attaques Man-in-the-Middle (MITM).
— Absence de restrictions d'accès : Aucun pare-feu ni règle de filtrage ne doit limiter l'accès aux services.

— Présence d'identifiants faibles : Un attaquant peut utiliser des attaques brute-force ou credential stuffing pour compromettre les comptes.

Impacts potentiels pour l'organisation en cas d'exploitation réussie

Une exploitation réussie de cette vulnérabilité peut entraîner un accès non autorisé aux informations confidentielles, mettant en danger la sécurité des données de l'organisation. Cela peut avoir plusieurs conséquences graves :

— **Atteinte à la confidentialité** : Vol ou fuite de données sensibles (données clients, informations internes, identifiants d'accès).
— **Perte de confiance** : Clients et partenaires peuvent douter de la fiabilité de l'organisation, affectant sa réputation.
— **Pertes financières** : Sanctions légales (RGPD, ISO 27001), coûts de remédiation et d'enquête, interruption des services impactant la productivité.
— **Risques opérationnels** : Un attaquant pourrait altérer, supprimer ou chiffrer des données (ransomware), entraînant une paralysie des activités.

0.1.3 Serveur Windows Server CVE-2023-23384

Conditions techniques d'exploitation

Pour qu'un attaquant puisse exploiter cette vulnérabilité, les conditions suivantes doivent être réunies :

— Présence de SQL Server vulnérable : Le serveur Windows doit héberger une version de Microsoft SQL Server affectée par cette vulnérabilité.
— Utilisation de l'outil SQLcmd : L'exploitation nécessite que l'outil SQLcmd soit utilisé sur le serveur. Cet outil est couramment employé pour exécuter des commandes T-SQL depuis la ligne de commande.
— Envoi de commandes malveillantes : L'attaquant doit pouvoir envoyer des commandes spécialement conçues via SQLcmd pour provoquer un dépassement de mémoire tampon, ce qui peut conduire à l'exécution de code arbitraire.

Impacts potentiels pour l'organisation en cas d'exploitation réussie

Une exploitation réussie de CVE-2023-23384 peut avoir des répercussions majeures sur l'organisation, combinant compromission des données, interruption des services, risques financiers et atteinte à la réputation.

0.1.4 Configuration non sécurisée de serveur Microsoft

Conditions techniques d'exploitation

Une configuration non sécurisée sur Windows Server, avec des comptes administratifs obsolètes non supprimés et une politique de mots de passe faible, expose le système à des attaques potentielles. L'exploitation repose sur plusieurs conditions :

1. Présence de comptes administratifs obsolètes
   — Comptes inactifs : Anciens comptes administrateurs laissés actifs sans surveillance.

- Absence de surveillance des connexions : Pas de journalisation des connexions sur ces comptes, permettant une exploitation furtive.

2. Politique de mots de passe faible
   - Mots de passe courts ou simples : Permet des attaques par brute-force ou rainbow tables.
   - Pas de rotation des mots de passe : Augmente le risque d'exploitation sur le long terme.
   - Absence d'authentification multifactorielle (MFA) : Rend plus facile la compromission des comptes.
3. Accès et exploitation possibles
   - Un attaquant ayant un accès au réseau peut tester les comptes obsolètes et deviner un mot de passe faible.
   - Exploitation des comptes inactifs via des outils comme Mimikatz pour récupérer des identifiants en mémoire.
   - Élévation de privilèges : Un compte compromis peut être utilisé pour obtenir un accès total au serveur.

Impacts potentiels pour l'organisation en cas d'exploitation réussie

L'exploitation d'une mauvaise gestion des comptes administratifs et d'une faible politique de mots de passe peut avoir des conséquences dévastatrices :

- **Perte de données**
- **Compromission totale du système**
- **Chiffrement ou suppression des fichiers critiques**
- **Pertes financières et légales**

0.2 Partie 2 : Proposition de gestion des correctifs

0.2.1 CVE-2022-22720

Correctif spécifique et téléchargement

- **Correctif :** Mise à jour d'Apache HTTP Server vers la version 2.4.53 ou ultérieure.
- **Téléchargement :** Disponible sur le site officiel d'Apache (http ://httpd.apache.org) et via les dépôts des distributions Linux (APT, YUM).

Procédure d'application

1. Sauvegarde du serveur et des configurations.
2. Mise à jour via le gestionnaire de paquets :
   - Debian/Ubuntu : sudo apt update sudo apt install apache2
3. Redémarrage du service : `sudo systemctl restart apache2`

Stratégie d'application du correctif

- **Méthode recommandée :** Manuelle
- **Raisons :**
  - Permet un contrôle total sur la mise à jour.
  - Évite les problèmes de compatibilité en effectuant des tests avant déploiement.
  - Réduit les risques d'interruption de service imprévue.

Tests des correctifs en préproduction

Procédure de test

1. Déploiement sur un serveur de test répliquant la production.
2. Validation fonctionnelle : Tester les fonctionnalités critiques du serveur web.
3. Analyse des journaux (`sudo journalctl -u apache2`) pour détecter d'éventuelles erreurs.
4. Tests de sécurité pour confirmer la correction de la vulnérabilité (ex : outils de scan de vulnérabilités).

Critères avant le déploiement en production

- Stabilité du serveur après la mise à jour.
- Compatibilité avec les applications existantes.
- Absence d'erreurs ou de régressions dans les logs.
- Confirmation que la vulnérabilité est corrigée.

0.2.2 Configuration non sécurisée

- Plusieurs services inutiles activés, tels que FTP et Telnet, et ports ouverts non justifiés (21 et 23).

Correctifs spécifiques et téléchargement

- **Correctif :** Désactivation des services inutiles (FTP, Telnet) et fermeture des ports non justifiés (21, 23).
- **Téléchargement :** Aucun logiciel requis, les correctifs sont appliqués via des commandes système.

Procédure d'application

1. Vérification des services actifs :
   - sudo systemctl list-units –type=service

FIGURE 0.2.1 – services actifs

2. Désactivation et suppression des services inutiles :
   - Désactiver et arrêter les services :
     - sudo systemctl stop vsftpd telnet
     - sudo systemctl disable vsftpd telnet

FIGURE 0.2.2 –

   - Désinstaller les paquets :
     - sudo apt remove –purge vsftpd telnetd
3. Fermeture des ports via UFW (Uncomplicated Firewall) :
   - sudo ufw deny 21
   - sudo ufw deny 23
   - sudo ufw enable

Stratégie d'application du correctif

- **Méthode recommandée :** Manuelle
- **Raisons :**
  - Permet de choisir précisément les services à désactiver en fonction des besoins.
  - Évite l'interruption involontaire de services essentiels.
  - Offre plus de flexibilité en cas d'adaptation des règles réseau.

Tests des correctifs en préproduction

Procédure de test

1. Vérifier que les services FTP et Telnet sont bien désactivés :

— sudo systemctl status vsftpd telnet

2. Confirmer que les ports 21 et 23 sont fermés :
   — sudo netstat -tulnp | grep -E "21|23"
3. Scanner les ports pour s'assurer qu'ils ne sont plus accessibles :
   — sudo nmap -p 21,23 localhost

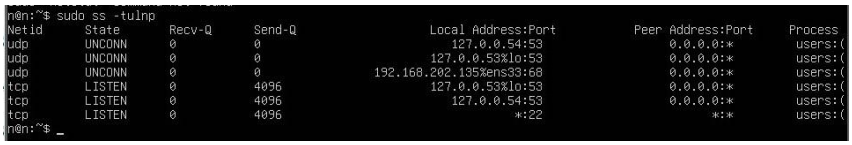

FIGURE 0.2.3 –

Critères avant le déploiement en production

— Aucun service FTP/Telnet actif.
— Ports 21 et 23 fermés et non accessibles.
— Aucune interruption des services critiques de l'organisation.
— Logs système sans erreurs liées à la désactivation des services.

0.2.3 CVE-2023-23384

Correctif spécifique et téléchargement

La vulnérabilité CVE-2023-23384 affecte Microsoft SQL Server, permettant une exécution de code à distance. Pour la corriger, Microsoft a publié des mises à jour de sécurité spécifiques.

— **Téléchargement :** Les correctifs sont disponibles via le Microsoft Security Response Center. Consultez la page dédiée à CVE-2023-23384 pour accéder aux liens de téléchargement appropriés.

Procédure d'application

1. Sauvegarde : Avant toute mise à jour, effectuez une sauvegarde complète de vos bases de données et configurations.
2. Installation : Téléchargez et appliquez le correctif correspondant à votre version de SQL Server en suivant les instructions fournies par Microsoft.
3. Redémarrage : Redémarrez les services SQL Server pour que les modifications prennent effet.

Stratégie recommandée pour l'application des correctifs

Une approche manuelle est préconisée pour ce type de mise à jour critique.

— **Raisons :**
   — Contrôle accru : Permet de superviser chaque étape du processus et de réagir rapidement en cas de problème.

— Tests préalables : Facilite la réalisation de tests en environnement de préproduction pour assurer la compatibilité et la stabilité.

Tests des correctifs en environnement de préproduction

Procédure de test

1. Déploiement en préproduction : Appliquez le correctif sur un serveur de test reproduisant fidèlement l'environnement de production.
2. Vérification fonctionnelle : Assurez-vous que toutes les fonctionnalités de SQL Server opèrent normalement après la mise à jour.
3. Analyse des journaux : Examinez les logs pour détecter d'éventuelles erreurs ou anomalies post-mise à jour.
4. Tests de charge : Soumettez le serveur à des charges similaires à celles de la production pour évaluer les performances.

Critères à vérifier avant le déploiement en production

— Stabilité : Le serveur doit fonctionner sans incidents après la mise à jour.
— Intégrité des données : Les bases de données doivent rester intactes et cohérentes.
— Compatibilité : Les applications dépendantes de SQL Server doivent continuer à fonctionner correctement.
— Sécurité : Confirmez que la vulnérabilité CVE-2023-23384 est effectivement corrigée.

Configuration non sécurisée

— Comptes administratifs obsolètes non supprimés et politique de mots de passe faible.

Correctifs spécifiques et téléchargement

— **Correctif :** Suppression des comptes administratifs obsolètes et renforcement de la politique de mots de passe.
— **Téléchargement :** Aucune mise à jour spécifique, les correctifs sont appliqués via des paramètres système et des scripts d'administration.

Procédure d'application

Désactiver les comptes inactifs :

— `Disable-ADAccount -Identity "NomUtilisateur"`

Supprimer les comptes non nécessaires :

— `Remove-ADUser -Identity "NomUtilisateur"`

Renforcement de la politique de mots de passe :

— Ouvrir `secpol.msc` et modifier :
  — Longueur minimale : 12 caractères.
  — Expiration : 90 jours.

— Historique : Conserver au moins 5 anciens mots de passe.
Appliquer les règles via PowerShell :
— `Set-ADDefaultDomainPasswordPolicy -Identity "NomDuDomaine" -MinPasswordLength 12 -MaxPasswordAge 90.00:00:00 -PasswordHistoryCount 5`

Stratégie d'application du correctif

— **Méthode recommandée :** Manuelle
— **Raisons :**
  — Prévenir les erreurs : Vérification manuelle avant suppression des comptes pour éviter toute suppression accidentelle.
  — Personnalisation : Ajuster la politique de mots de passe en fonction des besoins de l'organisation.

Tests des correctifs en préproduction

Procédure de test

1. Créer un environnement de test avec des comptes fictifs.
2. Tester la suppression des comptes et vérifier qu'aucun compte actif n'est impacté.
3. Vérifier la mise en place des nouvelles règles :
   — Essayer de créer un mot de passe trop court et s'assurer qu'il est refusé.
   — Tester l'expiration des mots de passe et la demande de renouvellement.
4. Analyser les journaux d'événements Windows (eventvwr.msc) pour détecter d'éventuels problèmes.

Critères avant le déploiement en production

— Aucun compte obsolète restant.
— Les utilisateurs actifs ne sont pas impactés.
— Les nouvelles règles de mot de passe sont bien appliquées.
— Aucune régression ni perte d'accès sur les comptes valides.

0.3 Partie 3 : Proposition de configurations sécurisées

Désactiver les services inutiles (ex : FTP, Telnet)

Serveur Ubuntu

J'ai installé les paquets de telnet et ftp et les ai mis en cours pour remédier aux problèmes de ports ouverts. Le suivant est l'exécution de la commande `systemctl list-units –type=service –state=running` qui nous montre que les processus telnet et ftp sont en cours.

Pour remédier aux problèmes de ports ouverts, j'ai arrêté les processus concernés comme le montrent les commandes suivantes :

Vérification de désactivation

Windows Server

Les commandes à exécuter dans Windows Server pour arrêter les FTP et Telnet :

— Identifier les services en cours d'exécution :

— Get-Service | Where-Object _.*Status* – *eq'Running'Dsactiverunservice*(*ex.Telnet, FTP*) :

— — Stop-Service -Name "TlntSvr" -Force

— Set-Service -Name "TlntSvr" -StartupType Disabled

— Stop-Service -Name "FTPSVC" -Force

— Set-Service -Name "FTPSVC" -StartupType Disabled

Vérifier qu'un service est bien désactivé :

```
Get-Service -Name "TlntSvr"
```

Fermer les ports ouverts non nécessaires

Ubuntu Linux

On vérifie les ports ouverts :

```
n@n:~$ sudo ss -tulnp
Netid      State      Recv-Q     Send-Q                Local Address:Port          Peer Address:Port     Process
udp        UNCONN     0          0                        127.0.0.54:53               0.0.0.0:*         users:(
udp        UNCONN     0          0                     127.0.0.53%lo:53               0.0.0.0:*         users:(
udp        UNCONN     0          0           192.168.202.135%ens33:68               0.0.0.0:*         users:(
tcp        LISTEN     0          4096                  127.0.0.53%lo:53               0.0.0.0:*         users:(
tcp        LISTEN     0          4096                     127.0.0.54:53               0.0.0.0:*         users:(
tcp        LISTEN     0          4096                             *:22                      *:*         users:(
n@n:~$ _
```

FIGURE 0.3.1 –

Fermer un port spécifique avec ufw (ex. 21 pour FTP, 23 pour Telnet) :

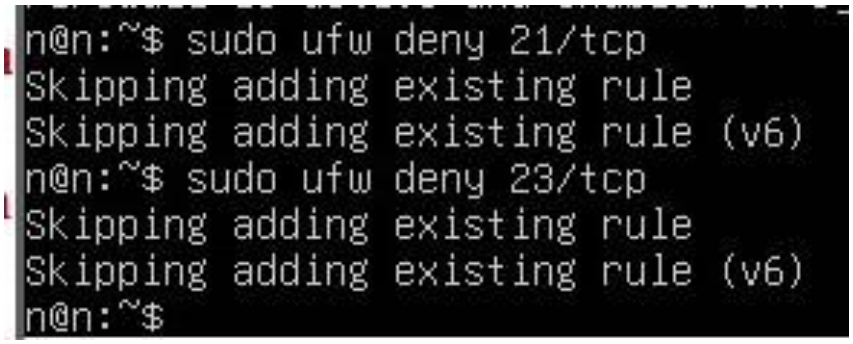

FIGURE 0.3.2 –

Exécution d'une fermeture générale, pour autoriser uniquement les ports nécessaires :

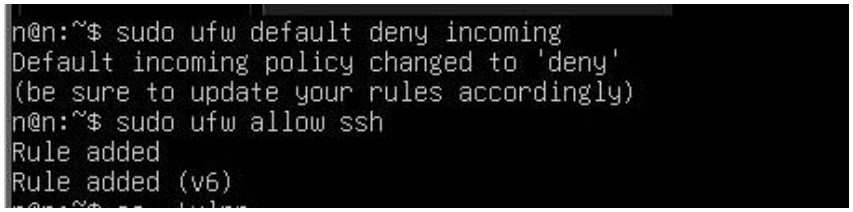

FIGURE 0.3.3 –

Windows Server

Vérifier les ports ouverts : Get-NetTCPConnection | Where-Object _.*State* –*eq*"*Listen*"

Fermer un port avec le pare-feu Windows (ex. 21 pour FTP, 23 pour Telnet) :

- New-NetFirewallRule -DisplayName "Block FTP" -Direction Inbound -LocalPort 21 -Protocol TCP -Action Block
- New-NetFirewallRule -DisplayName "Block Telnet" -Direction Inbound -LocalPort 23 -Protocol TCP -Action Block

Vérifier la règle appliquée : Get-NetFirewallRule -DisplayName "Block FTP"

Renforcer la gestion des comptes administratifs et la politique de mots de passe

Changer la configuration du fichier sudo nano /etc/security/pwquality.conf pour une politique de mot de passe forte.

0.3.1 Automatisation avec Ansible

Installation

—Ubuntu/Debian: sudo apt update && sudo apt install ansible -y

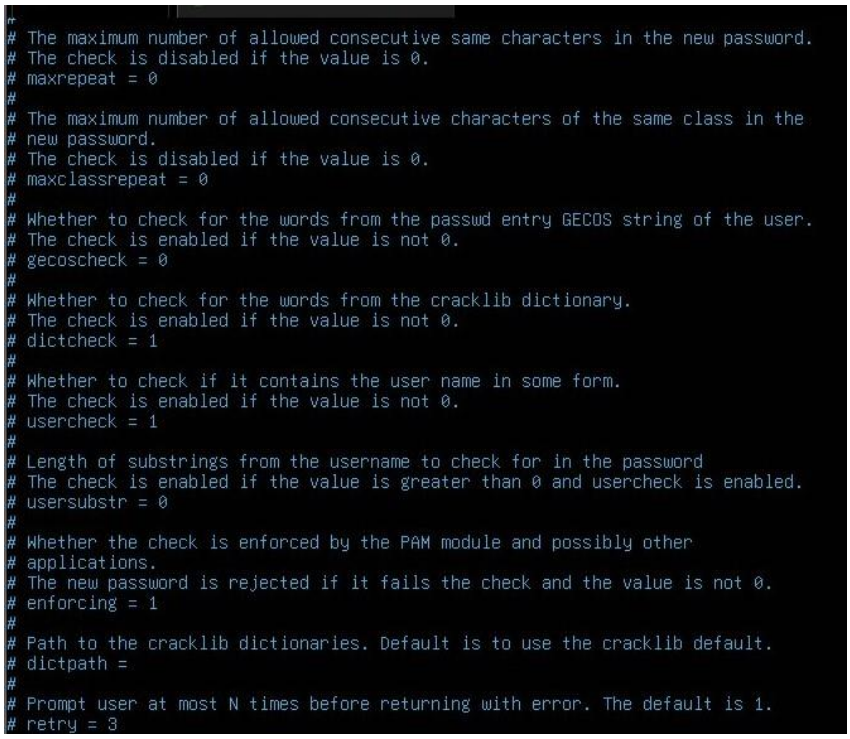

FIGURE 0.3.4 –

—CentOS/RHEL: sudo yum install ansible -y
—V'erification: ansible –version

Configuration

—Cr'eation d'un fichier d'inventaire inventory.ini listant les serveurs cibles.
—R'edaction d'un playbook secure.yml pour :
- Arr'eter des services inutiles (telnet, vsftpd)
- Bloquer les ports (21, 23)
- Appliquer une politique de mot de passe strict
- Configurer l'expiration des mots de passe

Exécution

—ansible-playbook -i inventory.ini secure.yml
—V'erification : ansible all -i inventory.ini -m shell -a "systemctl is-active vsftpd"

0.3.2 Automatisation avec Puppet

Installation

—Ubuntu/Debian: sudo apt update && sudo apt install puppet-agent -y
—CentOS/RHEL: sudo yum install puppet-agent -y

— V’erification: puppet –version

Configuration

— Cr’eation d’un module Puppet security avec un fichier init.pp incluant :
  - Arr’et des services inutiles
  - Blocage des ports
  - Application des r‘egles de s’ecurit’e sur les mots de passe
  - Expiration des mots de passe

Exécution

— puppet apply /etc/puppet/modules/security/manifests/init.pp
— V’erification : puppet resource service vsftpd

0.3.3 Automatisation avec Chef

Installation

— curl -L https://omnitruck.chef.io/install.sh | sudo bash
— V’erification: chef-client –version

Configuration

— G’en’eration d’un cookbook security
— R’edaction d’une recette default.rb pour :
  - Arr’eter des services inutiles
  - Bloquer des ports
  - Appliquer des politiques de mot de passe strictes
  - Configurer l’expiration des mots de passe

Exécution

— sudo chef-client –local-mode –runlist ’recipe[security]’
— V’erification : systemctl status vsftpd

0.4 Partie 4 : Vérification finale et audit de sécurité

Pour la vérification finale et l’audit de sécurité de nos systèmes après la mise en œuvre des actions correctives, On suit ces étapes:

0.4.1 Vérification des Correctifs

Ubuntu

— Vérification de la version Apache corrigée : [language=bash] apache2 -v

— Vérification de la désactivation de FTP et Telnet : [language=bash] sudo systemctl status vsftpd telnet
— Vérification des ports fermés : [language=bash] sudo netstat -tulnp | grep -E "21|23"

Windows Server

— Vérification de la version SQL Server : [language=PowerShell] sqlcmd -S <NomDuServeur> -Q "SELECT @@VERSION"
— Vérification des services désactivés : [language=PowerShell] Get-Service | Where-Object .*Status* – *eq*'*Running*'

0.4.2 Conformité aux Benchmarks CIS

— Utilisation de CIS-CAT : Télécharger et exécuter depuis https://www.cisecurity.org/cis-cat/CIS-CAT.
— Vérification des politiques de mots de passe : [language=bash] sudo cat /etc/login.defs sudo cat /etc/security/pwquality.conf
— Vérification des comptes inactifs sous Windows : [language=PowerShell] Get-ADUser -Filter * -Properties LastLogonDate | Where-Object .*LastLogonDate* – *lt*(*Get* – *Date*).*AddDays*(–90)

0.4.3 Outils et Techniques d'Audit

— Analyse des vulnérabilités : Utilisation de Nessus ou OpenVAS.
— Analyse des logs : [language=bash] sudo journalctl -u apache2
— Gestion de la configuration : Utilisation d'Ansible ou Puppet.

0.5 conclusion

Grâce à cette analyse, les vulnérabilités critiques des serveurs Linux et Windows ont été identifiées, évaluées et corrigées à l'aide de mesures adaptées. L'application des correctifs et des configurations sécurisées conformes aux CIS Benchmarks renforce la posture de sécurité de l'entreprise. Enfin, la mise en place d'un audit final garantit la conformité et l'efficacité des actions menées, réduisant ainsi les risques liés aux cybermenaces.