CTF-3 Walkthrough

Objectif: Trouver un flag caché dans un fichier secret en exploitant les permissions Linux et les mauvaises pratiques de configuration

---

Scénario

Vous êtes un pentester qui a obtenu un accès initial à un système Linux en tant qu'utilisateur non privilégié `player`. Votre mission est d'explorer le système, d'identifier les fichiers sensibles et de récupérer le flag caché dans les dossiers d'un autre utilisateur.

Ce challenge simule une situation réelle où :

- Des fichiers de configuration contiennent des informations sensibles
- Les permissions Linux ne sont pas correctement configurées
- Des mots de passe sont stockés en clair dans des fichiers accessibles

---

Reconnaissance

Étape 1: Démarrer le challenge

Lancez le conteneur Docker :

```
cd CTF/CTF-3-linux-permissions
docker-compose up -d --build
```

Entrez dans le conteneur :

```
docker exec -it ctf3-linux-permissions /bin/bash
```

Étape 2: Identifier votre contexte

À la connexion, vous verrez un message de bienvenue. Identifiez votre utilisateur :

```
whoami
# Sortie : player

id
# Sortie : uid=1001(player) gid=1001(player) groups=1001(player)
```

Observation : Vous êtes `player`, un utilisateur non privilégié sans accès sudo.

Étape 3: Explorer votre répertoire personnel

```
pwd
# Sortie : /home/player

ls -la
# Sortie :
# drwxr-xr-x 2 player player 4096 welcome.txt
# -rwxr-xr-x 1 player player 2048 helper.py
```

Lisez le fichier de bienvenue :

```
cat welcome.txt
```

Le fichier donne des indices :

- Les fichiers cachés commencent par un point (.)
- Il existe d'autres utilisateurs sur le système
- Vérifier les permissions des fichiers

Étape 4: Utiliser le script helper.py

```
python3 helper.py
```

Sortie :

```
╔═══════════════════════════════════════╗
║   CTF Linux Permissions -             ║
║   Linux System Exploration            ║
╚═══════════════════════════════════════╝


Your objective is to find the flag hidden somewhere in the Linux system.
Hint: the flag is in a hidden file...

 Useful Linux commands:
  • ls -la                              # List all files (including hidden ones)
  • cd /home                            # Go to the users directory
  • find / -name '*.txt' 2>/dev/null    # Search for all .txt files
  • grep -r 'password' /opt 2>/dev/null    # Search for the word 'password'
  • cat <file>                          # Display the contents of a file
  • su <username>                       # Switch user
```

Indices importants :

- Chercher des fichiers `.txt`
- Le mot "password" pourrait être intéressant
- La commande `su` pour changer d'utilisateur

Énumération système

Étape 5: Lister les utilisateurs du système

```
ls -la /home
# Sortie :
# drwxr-xr-x 2 admin  admin  4096 .
# drwxr-xr-x 2 player player 4096 .
```

Découverte : Il existe un utilisateur `admin` avec un répertoire personnel !

Essayez d'accéder au répertoire admin :

```
cd /home/admin
ls -la
# Sortie :
# drwxr-xr-x 2 admin admin 4096 .
# -rw------- 1 admin admin   17 .secret.txt
```

Observation critique :

- Un fichier `.secret.txt` existe (fichier caché !)
- Permissions `600` : seul `admin` peut lire/écrire
- Vous ne pouvez pas le lire directement

```
cat /home/admin/.secret.txt
# Sortie : Permission denied
```

Étape 6: Chercher des informations sensibles

Le helper suggère de chercher le mot "password". Essayons :

```
grep -r 'password' /opt 2>/dev/null
```

Résultat :

```
/opt/system/notes.txt:TODO: Change default admin password!
/opt/system/notes.txt:Current temporary password: SecureP@ss2026
```

Un fichier de configuration contient le mot de passe de l'administrateur !

Étape 7: Examiner le fichier de configuration

```
cat /opt/system/notes.txt
```

Contenu :

```
# System Configuration Notes

Date: 2026-01-01
Admin: admin

## Password Management
TODO: Change default admin password!
Current temporary password: SecureP@ss2026

REMINDER: This file should be removed in production!
```

Vulnérabilité identifiée :

- Mot de passe stocké en clair dans un fichier
- Permissions `644` : lisible par tous les utilisateurs
- Le fichier aurait dû être supprimé en production

Exploitation

Étape 8: Élévation de privilèges

Maintenant que nous avons le mot de passe admin, connectons-nous :

```
su admin
Password: SecureP@ss2026
```

Vous êtes maintenant `admin`.

```
whoami
# Sortie : admin

id
# Sortie : uid=1000(admin) gid=1000(admin) groups=1000(admin)
```

Étape 9: Récupérer le flag

En tant qu'admin, vous pouvez maintenant lire le fichier secret :

```
cd /home/admin
ls -la
# Sortie :
# -rw------- 1 admin admin 17 .secret.txt

cat .secret.txt
```

FLAG TROUVÉ : `CTF{affe46haf}`

Mesures de protection

1. Permissions strictes sur les fichiers sensibles

```
# Avant (vulnérable)
chmod 644 /opt/system/notes.txt

# Après (sécurisé)
chmod 600 /opt/system/notes.txt
chown root:root /opt/system/notes.txt
```

2. Ne jamais stocker de mots de passe en clair

Alternatives sécurisées :

a) Gestionnaire de secrets

```
# Utiliser HashiCorp Vault
vault kv put secret/admin password="SecureP@ss2026"

# Récupération sécurisée
vault kv get -field=password secret/admin
```

b) Variables d'environnement (avec précautions)

```
# Stocker dans un fichier .env avec permissions strictes
echo "ADMIN_PASSWORD=SecureP@ss2026" > .env
chmod 600 .env

# Charger dans l'environnement
source .env
```

c) Authentification par clé SSH

```
# Générer une paire de clés
ssh-keygen -t ed25519 -C "admin@system"

# Copier la clé publique
ssh-copy-id admin@server

# Désactiver l'authentification par mot de passe
echo "PasswordAuthentication no" >> /etc/ssh/sshd_config
```

3. Principe du moindre privilège

```
# Créer un groupe dédié pour les fichiers sensibles
groupadd secrets-readers
chgrp secrets-readers /opt/system/notes.txt
chmod 640 /opt/system/notes.txt

# Ajouter uniquement les utilisateurs autorisés
usermod -aG secrets-readers authorized_user
```

4. Audit et surveillance

```
# Configurer auditd pour surveiller l'accès aux fichiers sensibles
auditctl -w /opt/system/notes.txt -p rwa -k sensitive_file_access

# Vérifier les logs
ausearch -k sensitive_file_access
```

5. Nettoyage des fichiers temporaires

```
# Dockerfile amélioré
RUN echo 'Notes de configuration' > /opt/system/notes.txt && \
    chmod 600 /opt/system/notes.txt && \
    # Supprimer après utilisation
    rm -f /opt/system/notes.txt
```

Élévation de privilèges

Méthodes d'élévation :

1. **su** (Switch User) - Nécessite le mot de passe de l'utilisateur cible
2. **sudo** - Utilise votre propre mot de passe (si autorisé dans sudoers)
3. **Exploitation de binaires SUID** (voir CTF-4)
4. **Exploitation de vulnérabilités kernel**

Commandes utiles

Énumération

```
# Lister tous les fichiers (y compris cachés)
ls -la

# Trouver tous les fichiers .txt
find / -name "*.txt" 2>/dev/null

# Chercher un mot dans des fichiers
grep -r "password" /opt 2>/dev/null

# Trouver des fichiers modifiés récemment
find /home -type f -mtime -7
```

Permissions

```
# Afficher les permissions
ls -l fichier.txt

# Modifier les permissions (numérique)
chmod 600 fichier.txt

# Modifier les permissions (symbolique)
chmod u+rwx,go-rwx fichier.txt

# Changer le propriétaire
chown user:group fichier.txt
```

Gestion des utilisateurs

```
# Changer d'utilisateur
su username

# Voir l'utilisateur actuel
whoami

# Voir les groupes et UID/GID
id

# Lister les utilisateurs système
cat /etc/passwd

# Voir les utilisateurs connectés
who
```

Ressources complémentaires

- Linux File Permissions Explained
- OWASP - Sensitive Data Exposure
- CIS Benchmark - File Permissions
- GTFOBins - Unix Binaries Exploitation