ADMINISTRATION ET SECURITE SOUS LINUX

Armel KEUPONDJO

PLAN

- GESTION DES HOTES
- LES SERVICES
- LA SECURITE DES SERVICES

AUDITER SON SERVEUR

❑ Netstat

netstat, pour « **network statistics** », est une ligne de commande affichant des informations sur les connexions réseau, les tables de routage et un certain nombre de statistiques dont ceux des interfaces, sans oublier les connexions masquées, les membres multicast, et enfin, les messages netlink. La commande est disponible sous Unix (et ses dérivés dont Linux) et sous Windows NT compatibles.

Observation des sessions TCP établies, des ports ouverts.

netstat -a

netstat -tnp

netstat -ltnp

netstat -s

netstat -r

AUDITER SON SERVEUR

❑ Nmap

Nmap (“**Network Mapper**”) est un outil open source d’exploration réseau et d’audit de sécurité. Il a été conçu pour rapidement scanner de grands réseaux, mais il fonctionne aussi très bien sur une cible unique.

Nmap est généralement utilisé pour les audits de sécurité mais de nombreux gestionnaires de systèmes et de réseaux l’apprécient pour des tâches de routine comme les inventaires de réseau, la gestion des mises à jour planifiées ou la surveillance des hôtes et des services actifs.

AUDITER SON SERVEUR

❑ Nmap

L'état d'un port est soit :

- ✓ **ouvert (open)** : indique que l'application de la machine cible est à l'écoute de paquets/connexions sur ce port.
- ✓ **filtré (filtered)** : indique qu'un pare-feu, un dispositif de filtrage ou un autre obstacle réseau bloque ce port, empêchant ainsi Nmap de déterminer s'il s'agit d'un port ouvert ou fermé.
- ✓ **fermé (closed)** : n'ont pas d'application en écoute, bien qu'ils puissent quand même s'ouvrir n'importe quand.
- ✓ **ou non-filtré (unfiltered)** : les ports répondent aux paquets de tests (probes) de Nmap, mais Nmap ne peut déterminer s'ils sont ouverts ou fermés.

AUDITER SON SERVEUR

❑ Nmap

Quelques commandes utiles:

- ✓ Scan traditionnel en mode SYN (TCP): *nmap -sT -Pn 192.168.1.254*
- ✓ Scan furtif en mode SYN (TCP): *nmap -sS -Pn 192.168.1.254*
- ✓ Scan traditionnel en mode non SYN (UDP): *nmap -sU 192.168.1.254*
- ✓ ou non-filtré (unfiltered) : les ports répondent aux paquets de tests (probes) de Nmap, mais Nmap ne peut déterminer s'ils sont ouverts ou fermés.

AUDITER SON SERVEUR

- NOTES

UNE AIDE AVEC MAN VOUS DONNE LES INFORMATIONS SUR L'UTILISATIONS DE SES DEUX COMMANDES

SECURISER AVEC IPTABLES

❑ Introduction

Iptables est un utilitaire Linux puissant qui permet aux administrateurs système de configurer le pare-feu intégré du noyau. **Iptables** utilise un ensemble de règles pour déterminer comment filtrer le trafic réseau. Chaque règle précise quel type de trafic à filtrer et quelle action à prendre sur le trafic correspondant.

Dans ce module, nous aborderons quelques règles et commandes **Iptables** de base pour aider à sécuriser votre serveur. Par défaut, **Iptables** bloque tout le trafic entrant et autorise tout le trafic sortant. Ce n'est pas très sécurisé, nous devrons donc ajouter quelques règles pour rendre notre serveur plus sécurisé.

SECURISER AVEC IPTABLES

❑ Syntaxe de base

Avant de commencer, passons en revue la syntaxe de base pour iptables. La syntaxe générale pour **iptables** est la suivante :

Iptables -A <chain> -p <protocol> -s <source> -d <destination> -j <action>

SECURISER AVEC IPTABLES

❑ Syntaxe de base

où:

- <chain> est le nom de la chaîne (expliqué ci-dessous).
- <protocol> est le protocole du trafic (généralement TCP, UDP ou ICMP).
- <source> est l'adresse IP source.
- <destination> est l'adresse IP de destination.
- <action> est l'action à prendre (généralement ACCEPTER ou REJECTER).

SECURISER AVEC IPTABLES

❑ Syntaxe de base

chaînes

Les chaînes sont utilisées pour regrouper les règles **iptables** associées. Il y a trois chaînes intégrées :

- ✓ INPUT: pour le trafic entrant.
- ✓ OUTPUT: pour le trafic sortant.
- ✓ FORWARD : dans le cas de trafic acheminé d'un réseau à un autre.

SECURISER AVEC IPTABLES

❑ Syntaxe de base

Actions

Il y a deux actions principales que nous pouvons faire avec **iptables** : ACCEPT et DROP.

- ✓ ACCEPT: permet le trafic à travers.
- ✓ DROP: bloque la circulation.

SECURISER AVEC IPTABLES

❑ Commandes de base

➢ Lister les règles

iptables -L -v

est utilisé pour lister toutes les règles d'une chaîne.

-v: *est utilisée pour lister les règles avec la sortie verbeuse.*

➢ Ajouter des règles

Iptables –A : est utilisé pour ajouter une règle en fin de chaîne

Exple: *iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT*

SECURISER AVEC IPTABLES

❑ Commandes de base

➢ Ajouter des règles

iptables –A : est utilisé pour ajouter une règle en fin de chaîne

Exple: *iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT*

-I est utilisée pour ajouter une règle à la position spécifiée dans une chaîne.

Exple: *iptables* ***-I*** *INPUT* ***2*** *-s 192.168.1.0/24 -j ACCEPT*

-p est utilisée pour spécifier le protocole et l'option **- -dport** est utilisée pour spécifier le port de destination.

Exple: *iptables -A INPUT -p tcp - -dport* ***22*** *-j ACCEPT*

SECURISER AVEC IPTABLES

❑ Commandes de base

➢ Suppression de règles

iptables –D : est utilisé pour supprimer une règle à la position spécifiée dans une chaîne.

Exple: *iptables -D INPUT 2*

-F utilisée pour supprimer toutes les règles d'une chaîne.

Exple: *iptables -F INPUT*

-X utilisée pour supprimer une chaîne définie par l'utilisateur..

Exple: *iptables -X mychain*

-P est utilisée pour spécifier la stratégie par défaut pour une chaîne.

Exple: *iptables -P INPUT DROP*

SECURISER AVEC IPTABLES

- Commandes de base

  - Sauvegarder des règles

Iptables-save : ***est utilisé pour enregistrer les règles actuelles d'iptables.***

Exple: *iptables-save > /etc/iptables.rules*

iptables-restore: est utilisé pour restaurer les règles iptables sauvegardées.

Exple: *iptables-restore < /etc/iptables.rules*

SECURISER AVEC UFW

❑ Introduction

UFW, ou Uncomplicated Firewall, est une interface de gestion de pare-feu simplifiée qui masque la complexité des technologies de filtrage de paquets de niveau inférieur telles que **iptables.**

Si vous souhaitez commencer à sécuriser votre réseau, et vous n'êtes pas sûr de l'outil à utiliser, UFW peut être le bon choix pour vous.

UFW est installé par défaut sur Ubuntu. S'il a été désinstallé pour une raison quelconque, vous pouvez l'installer avec ***sudo apt install ufw.***

SECURISER AVEC UFW

❑ Mise en place des politiques par défaut

Si vous commencez tout juste à utiliser votre pare-feu, les premières règles à définir sont vos politiques par défaut. Ces règles contrôlent la manière de traiter le trafic qui ne correspond pas explicitement à d'autres règles. Par défaut, UFW est configuré pour refuser toutes les connexions entrantes et autoriser toutes les connexions sortantes. Cela signifie que toute personne essayant d'atteindre votre serveur ne pourra pas se connecter, tandis que toute application à l'intérieur du serveur pourra atteindre le monde extérieur.

Pour définir les valeurs par défaut utilisées par UFW, utilisez ces commandes :

sudo ufw default deny incoming

sudo ufw default allow outgoing

SECURISER AVEC UFW

❑ Mise en place des politiques par défaut

Ces commandes définissent les valeurs par défaut pour refuser les connexions entrantes et autoriser les connexions sortantes. Ces paramètres par défaut du pare-feu peuvent suffire pour un ordinateur personnel, mais les serveurs doivent généralement répondre aux demandes entrantes d’utilisateurs extérieurs.

SECURISER AVEC UFW

❑ Autoriser les connexions

Si nous activions notre pare-feu UFW maintenant, il refuserait toutes les connexions entrantes. Cela signifie que nous devrons créer des règles qui autorisent explicitement les connexions entrantes légitimes – connexions **SSH** ou **HTTP**, par exemple – si nous voulons que notre serveur réponde à ce type de demandes.

Pour configurer votre serveur afin d'autoriser les connexions SSH entrantes, vous pouvez utiliser cette commande :

sudo ufw allow ssh

Cela créera des règles de pare-feu qui autoriseront toutes les connexions sur le port 22 qui est le port que le démon SSH écoute par défaut. **UFW** sait quel port allow ssh désigne parce qu'il est listé comme un service dans le fichier /etc/services.

SECURISER AVEC UFW

❑ Autoriser les connexions

Cependant, nous pouvons réellement écrire la règle équivalente en spécifiant le port au lieu du nom du service. Par exemple, cette commande fonctionne de la même manière que celle ci-dessus :

sudo ufw allow 22

Si vous avez configuré votre démon SSH pour utiliser un port différent, vous devrez spécifier le port approprié. Par exemple, si votre serveur SSH écoute sur le port 2222 vous pouvez utiliser cette commande pour autoriser les connexions sur ce port :

sudo ufw allow 2222

Maintenant que votre pare-feu est configuré pour autoriser les connexions SSH entrantes, nous pouvons l'activer.

sudo ufw enable

SECURISER AVEC UFW

❑ Autoriser les connexions

Vous recevrez un avertissement qui indique que la commande peut perturber les connexions SSH existantes. Nous avons déjà mis en place une règle de pare-feu qui autorise les connexions SSH, donc nous pouvons continuer. Répondez à l'invite avec **y**.

Le pare-feu est maintenant actif. Exécutez la commande ***sudo ufw status verbose*** pour connaître les règles fixées actuellement.

Maintenant, vous devez autoriser toutes les autres connexions auxquelles votre serveur a besoin de répondre. Les connexions que vous devez autoriser dépendent de vos besoins spécifiques. Heureusement, vous savez déjà comment écrire des règles qui autorisent les connexions basées sur un nom de service ou un port ; nous l'avons déjà fait pour SSH sur le port 22.

SECURISER AVEC UFW

❑ Autoriser les connexions Adresses IP spécifiques

Lorsque vous travaillez avec **UFW**, vous pouvez également spécifier des adresses **IP**. Par exemple, si vous souhaitez autoriser les connexions à partir d'une adresse IP spécifique, comme une adresse IP professionnelle ou personnelle de 203.0.113.4 vous devez spécifier from puis l'adresse IP

sudo ufw allow from 203.0.113.4

Vous pouvez également spécifier un port spécifique auquel l'adresse IP est autorisée à vous connecter en ajoutant to any port suivi du numéro de port. Par exemple, Si vous souhaitez autoriser 203.0.113.4 à se connecter au port 22.

sudo ufw allow from 203.0.113.4 to any port 22

SECURISER AVEC UFW

❑ Autoriser les connexions réseaux/sous réseaux

Si vous souhaitez autoriser un sous-réseau d'adresses IP, vous pouvez le faire en utilisant la notation CIDR pour spécifier un masque de réseau. Par exemple, si vous souhaitez autoriser toutes les adresses IP allant de 203.0.113.1 à 203.0.113.254 vous pourriez utiliser cette commande :

sudo ufw allow from 203.0.113.0/24

De même, vous pouvez également spécifier le port de destination auquel le sous-réseau est autorisé à se connecter.

sudo ufw allow from 203.0.113.0/24 to any port 22

SECURISER AVEC UFW

❑ Connexion à une interface réseau spécifique

Si vous souhaitez créer une règle de pare-feu qui s'applique uniquement à une interface réseau spécifique, vous pouvez le faire en spécifiant « allow in on » suivi du nom de l'interface.

sudo ufw allow in on eth0 to any port 80

Ou, si vous voulez que votre serveur de base de données MySQL coute les connexions sur l'interface de réseau privé eth1 par exemple, vous pourriez utiliser cette commande :

sudo ufw allow in on eth0 to any port 3306

Cela permettrait à d'autres serveurs de votre réseau privé de se connecter à votre base de données MySQL.

Pour Refuser les connexions, utiliser les commandes décrites ci-dessus, en remplaçant allow par deny.

SECURISER AVEC UFW

❑ Suppression de règles

Savoir comment supprimer des règles de pare-feu est tout aussi important que de savoir comment les créer. Il existe deux façons différentes de spécifier les règles à supprimer : par le numéro de la règle ou par la règle elle-même (de la même façon que les règles ont été spécifiées lors de leur création).

❖ *Par numéro de règle*

Si vous utilisez le numéro de règle pour supprimer des règles de pare-feu, la première chose que vous voudrez faire est d'obtenir une liste de vos règles de pare-feu. La commande UFW status permet d'afficher des numéros à côté de chaque règle.

sudo ufw status numbered

Ensuite

sudo ufw delete numro-regle

SECURISER AVEC UFW

❑ Suppression de règles

Savoir comment supprimer des règles de pare-feu est tout aussi important que de savoir comment les créer. Il existe deux façons différentes de spécifier les règles à supprimer : par le numéro de la règle ou par la règle elle-même (de la même façon que les règles ont été spécifiées lors de leur création).

❖ *Par règle réelle*

L'alternative aux numéros de règle est de spécifier la règle réelle à supprimer. Par exemple, si vous voulez supprimer la règle allow http

sudo ufw delete allow http

ou

sudo ufw delete allow 80

SECURISER AVEC UFW

❑ Désactivation ou réinitialisation d'UFW (facultatif)

Si vous décidez que vous ne voulez pas utiliser UFW, vous pouvez le désactiver avec cette commande :

sudo ufw disable

Toutes les règles que vous avez créées avec UFW ne seront plus actives. Vous pourrez toujours exécuter la commande *sudo ufw enable* si vous devez l'activer plus tard.

Si vous avez déjà configuré des règles UFW mais que vous décidez de tout recommencer, vous pouvez utiliser la commande reset :

sudo ufw reset

Cela désactivera l'UFW et supprimera toutes les règles qui ont été définies précédemment.