Linux ?

Installation

Plusieurs saveurs de Linux sont distribuées avec un service SSH préinstallé qui est parfois déjà actif. D'autres distributions vous suggèreront de l'installer au moment de l'installation du système. Si le service doit être installé suite à l'installation du système, dépendamment de votre distribution, vous devrez vous référer au gestionnaire de packages ainsi qu'au répertoire logiciel de votre distribution.

Les commandes suivantes installeront et activeront SSH sur Debian et toutes les distributions découlant de Debian. Vous pouvez identifier ces distributions par le gestionnaire de packages "Aptitude" ou "apt".

```
# Installation
sudo apt update && apt install openssh-server
#
# Activation
sudo systemctl enable sshd && systemctl start sshd
```

Les commandes suivantes installeront et activeront SSH sur RHEL, Fedora ou toute autre distribution découlant de RedHat. Vous pouvez identifier ces distributions par le gestionnaire de packages "YUM" ou "DNF".

```
# Installation
sudo dnf install openssh-server
#
# Activation du service
sudo systemctl enable sshd && systemctl start sshd
```

Configuration

Le fichier de configuration du service/serveur est généralement le fichier /etc/ssh/sshd_config.

Configuration réseau

Les paramètres suivants retrouvés dans le fichier de configuration permettent de configurer les paramètres d'écoute du service sur le réseau. Le port par défaut est le port 22 et la configuration par défaut permet généralement autant aux adresses IPv4 que IPv6 de se connecter à n'importe

quelle interface réseau du serveur.

```
Port 22
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::
```

> Quoi que le port par défaut soit 22, le changer pour un port non standard évitera généralement les attaques par force brute.

Il est possible de limiter les connexions possibles à IPv4 ou IPv6 seulement en changeant le paramètre "AddressFamily" pour inet (IPv4) ou inet6 (IPv6). Les paramètres ListenAddress indiquent à quelles adresses du serveur il est possible d'établir une connexion SSH. Lorsque le paramètre est à 0.0.0.0 ou ::, il sera possible de se connecter au serveur par toutes ses interfaces réseau. Si une adresse IP y est mentionnée, seulement cette adresse pourra être utilisée pour établir une connexion au serveur.

Configuration d'accueil et tentatives d'authentification

Il est possible de configurer les actions qui seront entreprises lors d'une connexion ou une tentative de connexion au serveur. On peut par exemple, journaliser les tentatives d'authentification ou encore afficher une bannière à l'utilisateur au moment de la connexion. Ce genre de bannière est généralement utilisé pour indiquer des conditions d'utilisations ou présenter un avertissement comme quoi l'accès à ces systèmes sont restreints.

```
Banner [chemin vers un fichier de bannière]
SyslogFacility AUTH
```

Ici, le paramètre "**Banner**" doit faire référence à un fichier contenant la bannière d'accueil à afficher aux utilisateurs tentant de s'authentifier. Ce fichier pourrait contenir un message du genre à titre d'exemple.

```
Please don't hack us. Thx.
```

Ou encore présenter un message plus menaçant indiquant aussi aux utilisateurs que les tentatives d'authentification sont journalisées.

```
+---------------------------------------------------+
| UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE AND    |
| ATTACHED NETWORKS IS STRICTLY PROHIBITED.         |
| You must have explicit permission to access or    |
| configure this device. All activities performed on|
| this device may be logged or monitored without further|
```

```
| notice, and the resulting logs may be used as evidence|
| in court.                                              |
| Any unauthorized use of the system is unlawful, and    |
| may be subject to civil and/or criminal penalties.     |
+--------------------------------------------------------+
```

> Il est généralement pertinent de composer ces message en anglais pour que le plus d'acteurs tentant d'interagir avec le serveur puissent comprendre l'avertissement.

Le paramètre "**SyslogFacility**" peut être configuré avec les valeurs suivantes en fonction du niveau de sévérité des évènements qui doivent être journalisés :

- DAEMON
- USER
- AUTH
- LOCAL0 à LOCAL7 (Un niveau plus élevé inclus tous les niveaux plus bas. Ex. LOCAL7 incluera tous les messages, LOCAL1 incluera LOCAL0 et LOCAL1, etc.)
  - LOCAL0 sont pour les urgences seulement, le service est généralement non fonctionnel à ce moment
  - LOCAL1 est pour les alertes où une action doit être entreprise
  - LOCAL2 est pour les conditions critiques
  - LOCAL3 est pour les erreurs uniquement
  - LOCAL4 est pour les avertissements
  - LOCAL5 est pour les notifications
  - LOCAL6 est pour les journeaux informationels
  - LOCAL7 est pour les messages de débogage

> Il est généralement pertinent de journaliser l'ensemble des connexions, tant les connexions réussies que les connexions échouées. Ces journeaux permetteront d'identifier si le compte d'un utilisateur est compromis et exploité, si un utilisateur a accédé à un système et le nie ainsi que de détecter des tentatives d'authentification par force brute et prendre action en conséquence.

Configuration des permissions et méthodes d'authentification

Il existe deux méthodes principales d'authentification pour SSH : Authentification par clé publique (certificat) et authentification par nom d'usager et mot de passe. Par défaut, généralement tous les utilisateurs locaux du serveur ont l'autorisation de se connecter au serveur et idéalement, l'accès SSH par le compte "root" devrait être désactivée.

> Vérifiez lors du déploiment d'un nouveau serveur si le compte "root" a l'autorisation de se connecter au serveur. Ceci est généralement déconseillé car si ce compte est compromis, le

> serveur l'est aussi dans son entièreté. À moins d'un besoin spécifique, il est important de s'assurer de ceci.

L'accès par le compte "root" est géré par le paramètre "**PermitRootLogin**" qui peut avoir l'une des valeurs suivantes :

- yes

  Ce paramètre permet l'authentification tant par mot de passe que par clé publique

- no

  Ce paramètre restreint tout accès en SSH par le compte "root"

- prohibit-password (habituellement le paramètre par défaut)

  Ce paramètre permet l'authentification au compte root uniquement par une clé publique

- forced-commands-only

  Ce paramètre permet l'authentification au compte root uniquement par clé publique et restreint les commandes qu'il peut exécuter par le paramètre "**command**" qui devra aussi être spécifié dans le fichier "authorized_keys" situé dans le dossier ~/.ssh/ du compte root qui contient les clés publiques permettant l'authentification au compte en question.

Les deux autres paramètres pertinents sont "**PasswordAuthentication**" ainsi que "**PermitEmptyPasswords**" et s'expliquent d'eux même. Les valeurs possibles pour ces deux paramètres sont simplement "yes" et "no".

La gestion des autorisations pour les utilisateurs et groupes est possible avec les paramètres "**AllowUsers**", "**DenyUsers**", "**AllowGroups**" et "**DenyGroups**" suivis de soit une liste d'utilisateurs ou groupes séparés par un caractère d'espacement.

La plupart de ces paramètres peuvent être appliqués individuellements à des utilisateurs ou des groupes à l'aide du paramètre "**Match Group [liste de groupes]**" ou "**Match User [liste d'utilisateurs]**"

Configuration de services et sous-systèmes

SSH peut permettre beaucoup plus d'interactions entre un serveur et un client que d'ouvrir un shell de commande à distance. Il est possible d'établir des tunnels réseau, d'afficher une application graphique à distance (y compris un gestionnaire de fenêtres complet), de transférer des fichiers (par SCP ou par des sous-systèmes comme SFTP).

SCP est couvert plus en détail dans sa section du chapitre "clients" et permet de déposer et récupérer des fichiers à travers une session SSH. Ceci ne peut être désactivé à moins de forcer une

commande à exécuter à l'authentification SSH telle que "/bin/sh".

Les autres fonctionnalités toutefois sont paramétrables dans le fichier de configuration.

Le paramètre "AllowTcpForwarding" (ayant pour valeur "yes" ou "no") permet la redirection de ports à travers une connexion SSH. Ceci implique qu'un client peut mettre à la disposition du serveur une ressource située et accessible par le réseau du client, de mettre à sa disposition une ressource acccessible par le réseau du serveur ou même d'établir un proxy via une redirection de ports dynamique. Ceci est couvert plus en détail dans le chapitre client.

> Quoi que ceci soit dangereux car ceci permet à un attaquant de mettre à la disposition du serveur des ressources malveillantes de son côté ou de tenter d'accéder à des ressources distantes qui ne sont généralement pas accessibles à la ligne de commande (ex. page web de configuration d'un appareil), si un attaquant a accès à votre serveur et que l'exécution des commandes de l'utilisateur compromis permet d'accéder à ces ressources de toute façon, les systèmes distants seront aussi mis à la disposition d'un attaquant.

Le paramètre "**X11Forwarding**" (ayant à nouveau pour valeur "**yes**" ou "**no**") permet d'exécuter une application graphique sur un serveur distant et de l'afficher sur un "serveur" xorg du côté du client. Notez que ce paramètre fonctionne uniquement pour les applications exécutées en "X11" et ne fonctionnent généralement pas avec les applications exécutées en "Wayland".

Le(s) paramètres "**Subsystem**" permettent d'établir une connexion à différents services à travers une connexion SSH. Le sous-système sftp-server est généralement activé par défaut et nécessaire au fonctionnement de SCP2. Ce paramètre est composé d'une chaîne de caractères utilisé à la connexion SSH pour déterminer à quel exécutable la connexion sera établie.

La configuration suivante permettrait à un client d'établir une connexion à SFTP en précisant le sous-système à contacter lors de la connexion (ex. ssh -s sftp-server utilisateur@adresse.du.serveur)

```
Subsystem sftp /usr/lib/openssh/sftp-server
```

La configuration suivante permettrait à un client d'établir une connexion à un serveur IMAP (courriels) à travers une connexion SSH sécurisée en ajustant la commande du paragraphe précédent en conséquence du nom du sous-système

```
Subsystem imap /usr/sbin/imapd
```

Exemple typique d'un fichier de configuration sshd_config

Le détail du résultat des paramètres sont situés en commentaires (suivant #) dans l'exemple.

```
# Le serveur n'acceptera que les connexions IPv4 entrantes à l'adresse IP 192.168.0.10 au port TCP 20222
#
Port 20222
AddressFamily inet
ListenAddress 192.168.0.10

# Le contenu du fichier /etc/ssh/motd sera affiché à la connexion de l'utilisateur et les tentatives d'authentification seront journalisées
#
Banner /etc/ssh/motd
SyslogFacility AUTH

# Le compte root ne peut être utilisé pour s'authentifier en SSH, l'authentification par mot de passe est permise mais si un compte ne possède pas de mot de passe, il ne pourra pas être utilisé pour se connecter en SSH
#
PermitRootLogin no
PasswordAuthentication yes
PermitEmptyPasswords no

# Seul les groupes ssh_users et ssh_serviceaccounts pourront s'authentifier en SSH
#
AllowGroups ssh_users ssh_serviceaccounts

# Par défaut, tous les utilisateurs pourront établir des tunnels réseau et afficher à distance le contenu d'une application graphique
#
AllowTcpForwarding yes
X11Forwarding yes

# Il sera possible de transférer des fichiers par SFTP et SCP2 vers et depuis le serveur
#
Subsystem sftp /usr/lib/openssh/sftp-server

# Ces paramètres plus spécifiques s'appliqueront au groupe ssh_serviceaccounts
#
Match Group ssh_serviceaccounts
  AllowTcpForwarding no
  X11Forwarding no
```

```
  # Quoi que non couvert dans le document, ce paramètre détermine le nombre de tentatives de connexions permises (la valeur divisée par deux qui est 6 par défaut) avant qu'un échec d'authentification soit journalisé et que la session soit coupée. Un compte de service étant normalement utilisé pour des tâches automatisées, aucun échec d'authentification ne devrait avoir lieu
  MaxAuthTries 2

# Ces paramètres plus spécifiques s'appliqueront à l'utilisateur "bob"
#
Match user bob
  AllowTcpForwarding yes
  X11Forwarding no
  MaxAuthTries 80
  # Cette commande sera forcée à sa connexion et sera l'unique commande permise mais l'utilisateur pourrait utiliser ce serveur pour établir un tunnel réseau à condition qu'il soit dans le groupe "ssh_users".
  ForceCommand /bin/echo 'Pas de SSH pour toi bob.'
```

Revision #14
Created 2025-01-18 00:30:02 UTC by Alexandre Arsenault-Jetté
Updated 2025-01-18 06:07:52 UTC by Alexandre Arsenault-Jetté