ADMINISTRATION LINUX
ADMINISTRATION À DISTANCE

Description de Secure Shell

- Le paquetage OpenSSH fournit le protocole Secure Shell ou SSH dans Linux.
- Le protocole SSH permet aux systèmes de communiquer de manière chiffrée et sécurisée sur un canal d'un réseau non sécurisé.
- Utiliser la commande ssh pour créer une connexion sécurisée à un système distant, vous authentifier en tant qu'utilisateur spécifique et obtenir une session shell interactive sur le système distant.
- La commande ssh peut exécuter une session sur un système distant sans exécuter de shell interactif.

Exemples de Secure Shell

La commande ssh suivante vous connecte sur le serveur distant hosta en utilisant le même nom d'utilisateur que l'utilisateur local actuel. Dans cet exemple, le système distant vous invite à vous authentifier avec le mot de passe de l'utilisateur developer1.

```
[developer1@host ~]$ ssh hosta
developer1@hosta's password: redhat
...output omitted...
[developer1@hosta ~]$
```

```
[developer1@host ~]$ ssh developer2@hosta
developer2@hosta's password: shadowman
...output omitted...
[developer2@hosta ~]$
```

Utilisez la commande exit pour vous déconnecter du système distant.

```
[developer1@hosta ~]$ exit
logout
Connection to hosta closed.
[developer1@host ~]$
```

Identification des utilisateurs distants

La commande w affiche une liste des utilisateurs actuellement connectés au système. Elle affiche également l'emplacement du système distant et les commandes que l'utilisateur a exécutées.

```
[developer1@host ~]$ ssh developer1@hosta
developer1@hosta's password: redhat
[developer1@hosta ~]$ w
 16:13:38 up 36 min,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
developer2   pts/0    172.25.250.10    16:13    7:30   0.01s  0.01s -bash
developer1   pts/1    172.25.250.10    16:24    3.00s  0.01s  0.00s w
[developer2@hosta ~]$
```

L'utilisateur developer2 s'est connecté au système sur le pseudo-terminal 0 à 16:13 aujourd'hui à partir de l'hôte avec l'adresse IP 172.25.250.10 et a été inactif lors d'une invite du shell pendant sept minutes et trente secondes. La sortie montre également que l'utilisateur developer1 s'est connecté au système sur le pseudo-terminal 1 et a été inactif pendant les trois dernières secondes qui ont suivi l'exécution de la commande w.

Clés d'hôte SSH

- SSH sécurise les communications par chiffrement à clé publique.
- Lorsqu'un client SSH se connecte à un serveur SSH, ce dernier envoie une copie de sa clé publique au client avant qu'il ne se connecte.
- Cette clé permet de configurer le chiffrement sécurisé du canal de communication et d'authentifier le système du client.

- Lorsqu'un utilisateur utilise la commande ssh pour se connecter à un serveur SSH, la commande recherche une copie de la clé publique du serveur dans ses fichiers hôtes locaux connus.
- La clé peut être préconfigurée dans le fichier /etc/ssh/ssh_known_hosts, ou l'utilisateur peut avoir le fichier ~/.ssh/known_hosts qui contient la clé dans son répertoire personnel.

- Si le client possède une copie de la clé, la commande ssh compare la clé des fichiers hôtes connus de ce serveur à celle qu'il a reçue. Si les clés ne correspondent pas, le client suppose que le trafic réseau vers le serveur est compromis, et demande à l'utilisateur de confirmer s'il souhaite poursuivre la connexion.

La commande ssh demande une confirmation de connexion si le client ne possède pas de copie de la clé publique dans ses fichiers hôtes connus. La copie de la clé publique est enregistrée dans le fichier ~/.ssh/known_hosts pour confirmer automatiquement l'identité du serveur ultérieurement.

```
[developer1@host ~]$ ssh hostb
The authenticity of host 'hosta (172.25.250.12)' can't be established.
ECDSA key fingerprint is SHA256:qaS0PToLrqlCO2XGklA0iY7CaP7aPKimerDoaUkv720.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hostb,172.25.250.12' (ECDSA) to the list of known hosts.
developer1@hostb's password: redhat
...output omitted...
[developer1@hostb ~]$
```

```
[developer1@host ~]$ cat ~/.ssh/known_hosts
hosta,172.25.250.11 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOsEi0e+FlaNT6jul8Ag5Nj+RViZl0yE2w6iYUr+1fPtOIF0EaOgFZ1LXM37VFTx
dgFxHS3D5Whn1fb+68zf8+w=
```

Configuration de l'authentification par clé SSH

- Un accès sans mot de passe aux serveurs SSH
- L’authentification par clé doit être activée sur la destination
- Générer une paire de fichiers de clés liés au chiffrement.
- Une clé est privée et détenue uniquement par vous, tandis que la seconde est votre clé publique connexe qui n'est pas secrète.
- La clé privée correspond aux informations d'identification d'authentification et doit être stockée en toute sécurité.
- La clé publique est copiée sur votre compte sur les serveurs auxquels vous accéderez à distance et vérifie votre utilisation de votre clé privée.

Génération de clés SSH

la commande ssh-keygen crée une paire de clés. Par défaut, la commande ssh-keygen enregistre vos clés privées et publiques dans les fichiers ~/.ssh/id_rsa et ~/.ssh/id_rsa.pub, mais vous pouvez spécifier un nom différent.

```
[user@host ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): Enter
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:vxutUNPio3QDCyvkYm1 user@host.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|   .     .       |
|  o o     o      |
| . = o   o .     |
|  o + = S E .    |
| ..O o + * +     |
|.+% O . + B .    |
|=*oO . . + *     |
|++.     . +.     |
+----[SHA256]-----+
```

L'option -f de la commande ssh-keygen spécifie les fichiers dans lesquels enregistrer les clés.

Partage de la clé publique

- Pour configurer votre compte distant pour l'accès, copiez votre clé publique sur le système distant.
- La commande ssh-copy-id copie la clé publique de la paire de clés SSH vers le système distant.
- Vous pouvez spécifier une clé publique spécifique avec la commande ssh-copy-id ou utiliser le fichier ~/.ssh/id_rsa.pub par défaut.

```
[user@host ~]$ ssh-copy-id -i .ssh/key-with-pass.pub user@remotehost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
user@remotehost's password: redhat
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user@remotehost'"
and check to make sure that only the key(s) you wanted were added.
```

Partage de la clé privé

ssh-copy-id utilisateur_distant@adresse_IP_distante

Un message vous demande le mot de passe de la machine distante. Une fois l'authentification réussie, la clé publique SSH générée sera ajoutée au fichier authorized_keys de la machine distante.

Personnalisation de la configuration du service OpenSSH

- Le démon sshd fournit le service OpenSSH. Vous pouvez configurer le service en modifiant le fichier /etc/ssh/sshd_config.
- Pour apporter des modifications pour renforcer la sécurité de votre système.
- Interdire la connexion directe à distance au compte root, et peut-être interdire l'authentification par mot de passe (en faveur de l'authentification par clé privée SSH).

Interdiction de connexion du super utilisateur

Le serveur OpenSSH utilise les paramètres de configuration PermitRootLogin dans le fichier /etc/ssh/sshd_config pour autoriser ou interdire la connexion au système en tant que root aux utilisateurs.

```
PermitRootLogin yes
```

- Pour empêcher l'authentification par mot de passe mais autoriser l'authentification par clé privée pour root, définissez le paramètre PermitRootLogin sur without-password.

Interdiction de l'authentification par mot de passe pour SSH

- Ne pas autoriser les connexions à la ligne de commande à distance que par clé privée présente divers avantages :

- Les attaquants ne peuvent pas utiliser d'attaques en devinant des mots de passe pour pénétrer à distance dans des comptes connus sur le système.

- Avec les clés privées protégées par phrase de passe, un attaquant a besoin à la fois de la phrase de passe et d'une copie de la clé privée. Avec les mots de passe, un attaquant a juste besoin du mot de passe.

- En utilisant les clés privées protégées par phrase de passe avec ssh-agent, la phrase de passe est saisie et exposée moins souvent, et la connexion est plus pratique pour l'utilisateur.

```
PasswordAuthentication yes
```

Résumé

La commande ssh permet aux utilisateurs d'accéder en toute sécurité aux systèmes distants avec le protocole SSH.

Un système client stocke les identités de serveurs distants dans les fichiers ~/.ssh/known_hosts et /etc/ssh/ssh_known_hosts.

SSH prend en charge l'authentification par mot de passe et par clé.

La commande ssh-keygen génère une paire de clés SSH pour l'authentification. La commande ssh-copy-id exporte la clé publique vers des systèmes distants.

Le service sshd implémente le protocole SSH dans les systèmes Red Hat Enterprise Linux.

Configurez les paramètres SSH avancés dans le fichier de configuration /etc/ssh/sshd_config.

Il est recommandé de configurer sshd pour désactiver les connexions à distance en tant que root et d'exiger une authentification par clé publique plutôt que par mot de passe.