Création automatique et gestion de la clé SSH de l'utilisateur shinken

Sommaire

Création automatique lors de l'installation

Lors de l'installation de Shinken Enterprise sur un serveur Linux, l'installation créé automatiquement une clé SSH pour l'utilisateur shinken.

Chemin clé privée	/var/lib/shinken/.ssh/id_rsa
Chemin clé publique	/var/lib/shinken/.ssh/id_rsa.pub
Chiffrement	RSA
Taille de la clé	2048

C'est cette clé qui sera utilisée par défaut pour les connexions suivantes :

- Les checks linux basés sur SSH.
- La sécurisation des connexions MongoDB des différents démons et modules.

Connexion distante et déploiement des clés SSH

Principe des connexions SSH

Le SSH (*Secure SHell*) est un protocole de connexion sécurisé, basé sur un échange de clés privées et clés publiques. Il s'agit d'un chiffrement asymétrique.

Un utilisateur local (*qui sera dans notre cas* ***l'utilisateur shinken***) utilise sa clé privée pour se connecter à un serveur distant avec un compte distant, qui a autorisé la clé publique de cet utilisateur.

Dans le cas d'une connexion vers un serveur démon ou module Shinken, il faudra utiliser comme utilisateur distant celui qui est créé par défaut : shinken

Utilisation et génération des clés SSH

L'utilisateur "shinken" dispose déjà d'une clé SSH qui pourra être utilisée pour se connecter à divers serveurs tels que :

- D'autres serveurs Shinken
- Le même serveur Shinken pour utiliser un tunnel SSH pour la base MongoDB
- Un équipement supervisé

Vous pouvez être amené à générer de nouvelles clés :

- Pour différencier des connexions entre différents clients
- Dans le cas d'un cluster Mongo : l'utilisateur "root" doit pouvoir se connecter aux différents nœuds afin de garantir la mise à jour de ces serveurs
- Dans le cas où votre clé a été corrompue/volée

La génération d'une nouvelle clé se réalise avec l'utilisateur qui va utiliser la clé SSH (*dans cet exemple, il s'agit de l'utilisateur shinken*) :

```
[root@shinken-server ~]# su - shinken
[shinken@shinken-poller ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/shinken/.ssh/id_rsa):
/var/lib/shinken/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/shinken/.ssh/id_rsa.
Your public key has been saved in /var/lib/shinken/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Dti4LZ9+IGe6NWBbIFRjpKed4g3zzYfTx/jDfJ9iu7E shinken@shinken-poller
The key's randomart image is:
+---[RSA 2048]----+
|  .o=            |
| . o .           |
|  o o            |
|   = *           |
|  = B + S        |
| . B.O+= o       |
|  . *=O.=oo .    |
|    .+ =.o+ +o . |
|    .o+.  .+E=o  |
+----[SHA256]-----+
```

Lors de la génération d'une clé, alors qu'une clé existe déjà, il vous sera demander d'écraser la clé précédente.

Si vous ne souhaitez pas écraser la clé, mais en rajouter une :

- Lors de la génération de la clé, utiliser l'option "-f ~/.ssh/nom_de_la_nouvelle_cle" ;
- Pour l'utiliser en ligne de commande, pensez à utiliser l'option "-i chemin_de_la_clé" ;
- Celle-ci devra être spécifiée dans l'hôte/modèle d'hôte (*Ex. : linux*) ;
- Penser à renseigner le bon chemin dans les fichiers de configuration Shinken.

Autorisation d'un serveur Shinken vers un autre serveur Shinken

Dans le cas d'une installation de base

Dans le cas d'une connexion vers une base Mongo, ou d'un serveur shinken vers un autre, vous pouvez utiliser l'utilisateur "shinken" pour établir les tunnels SSH.

La méthode est la même, que le serveur MongoDB soit en local (*127.0.0.1 ou l'adresse du serveur*) ou soit un serveur distant.

L'utilisateur "shinken" ne dispose pas de mot de passe. Il n'est donc pas possible de passer par la commande "ssh-copy-id", il faut utiliser le compte root pour cela.

```
[root@shinken-server ~]# cat /var/lib/shinken/.ssh/id_rsa.pub | \
ssh root@groy-dev-02-08 \
"tee -a /var/lib/shinken/.ssh/authozied_keys"
```

Cela aura pour effet :

- D'afficher la clé publique de l'utilisateur shinken dans la sortie standard ;
- Puis de se connecter en SSH au serveur distant avec le compte "root" (*Si l'utilisateur "root" n'a pas sa clé publique déjà autorisée sur le serveur distant, le mot de passe "root" du serveur distant vous sera demandé*)
- Et enfin d'utiliser la clé présente dans la sortie standard pour ajouter la clé publique dans les clés autorisée pour se connecter à l'utilisateur "shinken"

Dans le cas d'un cluster Mongo

Pour les clusters Mongo, l'utilisateur "root" du serveur Shinken Central **doit accéder à tous les nœuds du cluster Mongo** avec les droits "root", afin de garantir la mise à jour de tous les nœuds.

Si votre utilisateur "root" ne dispose pas de clé SSh, générez une nouvelle clé SSH

Depuis le serveur Shinken central utilisez le compte "root" pour copier votre clé publique (*à faire avec tous les serveurs constituant le cluster Mongo*) :

```
[root@shinken-server ~]# ssh-copy-id root@AdressServeurMongo
The authenticity of host 'AdressServeurMongo(AdressServeurMongo)' can't be established.
RSA key fingerprint is 00:ff:ee:dd:cc:bb:aa:d6:d3:79:1d:f6:93:47:80:27.
Are you sure you want to continue connecting (yes/no)? yes
root@AdressServeurMongo's password:  XXXXXXXX
 Now try logging into the machine, with "ssh 'AdressServeurDistant'", and check in:
  .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

Autorisation d'un serveur Shinken vers un serveur tiers (pour le superviser)

Pour autoriser une clé d'un utilisateur "shinken" vers un serveur distant, il faut se connecter sur le serveur Shinken en tant que l'utilisateur "shinken" et copier la clé SSH vers le serveur distant**.**

Voici les paramètres utilisés de ce serveur dans cet exemple :

- Adresse : **AdressServeurDistant**
- Utilisateur **: UtilisateurDistant**
- Mot de passe : **XXXXXXXX** (*si vous n'avez pas le mot de passe de l'utilisateur distant, mais celui de l'utilisateur root, utiliser la méthode décrite dans le chapitre suivant :* Autorisation d'un serveur démon vers un autre serveur démon) en remplaçant l'utilisateur "shinken" par votre utilisateur

```
[root@shinken-server ~]# su - shinken
[shinken@shinken-poller ~]#  ssh-copy-id  AdressServeurDistant
The authenticity of host 'AdressServeurDistant (AdressServeurDistant)' can't be established.
RSA key fingerprint is 00:ff:ee:dd:cc:bb:aa:d6:d3:79:1d:f6:93:47:80:27.
Are you sure you want to continue connecting (yes/no)? yes
shinken@remote_host's password:  XXXXXXXX
 Now try logging into the machine, with "ssh 'AdressServeurDistant'", and check in:
  .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

Ceci aura pour effet de copier le contenu de la clé publique **/var/lib/shinken/.ssh/id_rsa.pub** dans le fichier **~/.ssh/authorized_keys** de l'utilisateur du serveur distant.

Pour superviser un élément par SSH, il est conseillé de prendre une clé privée de l'utilisateur shinken, car c'est l'utilisateur qui sera utilisé par défaut par le Poller pour lancer les checks.

Autorisation du service Grafana vers un serveur Shinken

Si vous utilisez le service Grafana, celui-ci à besoin d'accéder à la base MongoDB. La mise en place des clés SSH dans ce cadre est expliqué dans cette page : Grafana - v8.3.2

Test de connexion

Pour tester le déploiement d'une clé SSH il faut pouvoir se connecter depuis l'utilisateur qui dispose de la clé privée vers l'utilisateur sur le serveur distant. La connexion doit s'établir avec succès et sans authentification. Voici un exemple de connexion depuis l'utilisateur "shinken" vers l'utilisateur "shinken" d'un serveur distant :

```
[root@ServerShinken~]# su - shinken
[shinken@ServerShinken~]# ssh shinken@AdresseServerDistant -i /var/lib/shinken/.ssh/id_rsa
Last login: Tue Oct 19 15:53:19 2021 from ServerShinken
[shinken@AdresseServerDistant]#
```

L'option **-i** de la commande SSH permet de spécifier la clé à utiliser. Si l'option n'est pas utilisée, alors SSH essayera d'utiliser les chemins suivants :

- ~/.ssh/id_rsa
- ~/.ssh/id_dsa
- ~/.ssh/id_ecdsa
- ~/.ssh/id_ed25519

Lors de la première connexion à un serveur, la commande SSH vous affiche l'empreinte de la clé SSH, vous permettant de vérifier auprès de l'administrateur de ce serveur que c'est bien la bonne clé SSH qui est utilisée pour chiffrer les communications depuis le serveur distant vers le serveur local.

Une fois cette empreinte enregistrée, cela ne vous sera plus redemandé.

Bonnes pratiques

Dans le cas d'une installation Shinken dédié

Si vous disposez de plusieurs serveurs Shinken et que vous supervisez uniquement votre infrastructure, il est possible :

- d'avoir une clé SSH pour l'utilisateur "root" sur le serveur central qui sera autorisé sur tous les serveurs Shinken.
  - La clé publique devra être copiée sur tous les serveurs Shinken, pour l'utilisateur "root".
  - C'est obligatoire dans le cas d'un cluster Mongo pour pouvoir effectuer les mises à jour, lors d'une installation ou d'une mise à jour.
  - Cela facilite la configuration, la maintenance et la mise à jour.
- d'avoir une seule clé SSH sur l'utilisateur shinken, dans le but de l'utiliser sur toutes les machines supervisées par SSH.
  - La clé privée pourra être copiée sur tous les Pollers : une seule et même clé publique à autoriser sur tous les équipements à superviser.
  - La clé publique sera à copier sur tous les équipements.
  - Pour les tunnels SSH de connexion à Mongo.

Dans le cas d'une installation Shinken multi-clients

Si votre installation vous permet de superviser plusieurs clients, il est possible d'utiliser plusieurs clés SSH afin de séparer les connexions :

- d'avoir une clé SSH pour l'utilisateur "root" sur le serveur central qui sera autorisé sur tous les serveurs Shinken.
  - La clé publique devra être copiée sur tous les serveurs Shinken, pour l'utilisateur "root".
  - C'est obligatoire dans le cas d'un cluster Mongo pour pouvoir effectuer les mises à jour, lors d'une installation ou d'une mise à jour.
  - Cela facilite la configuration, la maintenance et la mise à jour.
- d'avoir une clé SSH sur l'utilisateur shinken
  - Pour les checks de supervision shinken (*sup de sup*)
  - Pour les tunnels SSH de connexion à Mongo
- Une clé par client
  - La clé privée pourra être copié sur tous les Pollers qui supervisent les équipements de ce client une seule clé publique à copier sur les équipements à superviser.
  - Cette clé privée sera à renseigner dans les modèles / hôtes de la configuration.

Cela permet de cloisonner les connexions entre votre infrastructure et vos clients. Cela permet aussi de pouvoir changer de clé SSH en cas de vol ou de corruption en limitant les impacts sur les autres clients.