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 mongo des différents démons et modules.

Connexion distante, et déploiement des clés SSH pour se connecter

Principe des connexions SSH

Les connexions SSH se basent sur l'utilisation de clés: un utilisateur local (*qui sera dans notre cas de* ***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 notre utilisateur shinken.

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

Autoriser la clé d'un utilisateur à se connecter

Autorisation de la clé

Pour autoriser une clé d'un utilisateur shinken vers un serveur distant, il faut se connecter sur le serveur source en tant que l'utilisateur "shinken" et copier la clé SSH vers le serveur **distant (utilisons ServerDistantIP pour symbolisé son adresse de connexion)**:

```
[root@shinken-server ~]# su - shinken
[shinken@shinken-poller ~]#  ssh-copy-id  ServerDistantIP
The authenticity of host '192.168.1.19 (192.168.1.19)' 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: XXXXXXXXXX
Now try logging into the machine, with "ssh '192.168.1.19'", and check in:
  .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.

ssh shinken@distant -i .ssh/id_rsa
```

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.

Test de connexion

Pour tester le bon déploiement de la clé SSH, il faut lancer depuis le serveur source:

```
[root@shinken ~]# su - shinken
[shinken@shinken ~]# ssh shinken@ServerDistantIP -i /var/lib/shinken/.ssh/id_rsa
```

La connexion doit s'établir avec succès.

Cas de SSH pour la connexion entre Graphite et la base MongoDB (pour utilisation de l'outil Grafana)

Dans le cas où on utilise l'outil Grafana, Graphite va avoir besoin de se connecter à la base MongoDB.

- Il est conseillé de protéger la connexion dans un tunnel SSH. Graphite étant hébergé par le service apache, il n'a donc pas accès au répertoire **/var/lib/shinken** et donc pas accès à la clé SSH **/var/lib/shinken/.ssh/id_rsa**.
- Pour plus de détail sur la connexion de Grafana avec Shinken, allez voir la page Grafana.

```
cp /var/lib/shinken/.ssh/id_rsa /opt/graphite/conf/id_rsa
chown apache:apache /opt/graphite/conf/id_rsa
```

Attention: un lien symbolique entre les deux fichiers ne fonctionnera pas, car apache n'a pas les droits d'aller lire le fichier originel.