Network Discovery

Contexte

Pour stocker ses données, Shinken Entreprise utilise le système de base de données MongoDB.

Par défaut, un paramètre permet de n'autoriser que les connexions à destination de sa boucle locale 127.0.0.1 (Voir le chapitre sécurisation via le paramètre bind_ip).

Cependant, si vous souhaitez autoriser MongoDB à écouter les requêtes à destination de ses autres interfaces locales (afin que d'autres serveurs puissent se connecter sur la base MongoDB par exemple), nous vous conseillons alors de créer des règles Iptables afin de n'autoriser uniquement ces serveurs spécifiques. (Voir le chapitre sécurisation via iptables).

Sécurisation via le paramètre bind_ip

Lors de l'installation de Shinken, MongoDB est installé et le paramétrage par défaut est dans /etc/mongod.conf.

Par défaut, le paramètre bind_ip permettant de spécifier l'interface d'écoute de MongoDB est à 127.0.0.1 :

```
# Listen to local interface only. Comment out to listen on all interfaces.
bind_ip=127.0.0.1
```

Dans ce cas, MongoDB ne répondra qu'aux requêtes envoyées sur son interface locale, c'est à dire uniquement les requêtes dont l'origine est le serveur MongoDB lui même. Aucun autre serveur ne pourra se connecter directement à la base MongoDB.

Cette sécurisation est très bonne, mais il se peut que vous ayez besoin que vos démons Shinken accèdent à la base MongoDB alors que ces derniers sont sur des serveurs distants (dans une architecture distribuée par exemple, ou encore si vous souhaitez dédier un serveur pour vos bases).

Dans ce cas, vous aurez besoin de commenter la ligne "bind_ip" ou de spécifier une interface autre que l'interface locale. Si vous faites cela, n'importe qui pourra alors passer des requêtes MongoDB à destination du serveur hébergeant les bases de données. Pour ne pas ouvrir cette faille de sécurité, il est **indispensable de sécuriser vos connexions** via des règles de Firewall ou des connexions SSH.

Dans le cas de la rétention de données basé sur MongoDB, il est possible d'utiliser un tunnel SSH pour sécuriser la connexion entre les serveurs Shinken et le serveur MongoDB. Pour cela, voir la page suivante : Rétention Mongodb

Sécurisation Via iptables

Voyons comment sécuriser vos connexions via les services "iptables" de Linux.

Activation chkconfig

Nous vous conseillons tout d'abord d'activer iptables comme service afin que ce dernier se charge dès le démarrage du serveur hébergeant MongoDB :

```
chkconfig iptables on
```

Autorisation des serveurs spécifiques

Il faut maintenant écrire nos règles de sécurisation (ACL) via les commandes iptables.

On autorise tout d'abord les connexions (entrantes et sortantes) avec les serveurs ayant comme IP x.x.x.x et y.y.y.y.y (pour notre exemple, avec por par défaut MongoDB 27017 - à adapter à votre environnement) :

```
iptables -A INPUT -s 127.0.0.1,x.x.x.x,y.y.y.y.y -p tcp --destination-port 27017 -m state --state NEW,
ESTABLISHED -j ACCEPT
iptables -A OUTPUT -d 127.0.0.1,x.x.x.x,y.y.y.y.y -p tcp --source-port 27017 -m state --state ESTABLISHED -j
ACCEPT
```

Info : vous pouvez utiliser les notations CIDR, exemple : 10.10.10.10/24 ou 10.10.10.10/255.255.255.0

Blocage des flux réseaux

Nous bloquons maintenant **explicitement** toutes les connexions (entrantes et sortantes) sur le port 27017 (port par défaut de MongoDB - attention, a adapter si vous avez changer le port) :

```
iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 27017 -j DROP
iptables -A OUTPUT -p tcp -d 0.0.0.0/0 --sport 27017 -j DROP
```

Vous pouvez également bloquer tous les autres flux avec "iptables -P INPUT DROP" et "iptables -P OUTPUT DROP" (attention tout de même aux autres applications de votre serveur qui nécessitent peut être des accès réseaux également).

Persistance de vos règles

Il s'agit maintenant de sauvegarder votre configuration afin qu'elle ne se réinitialise pas au redémarrage du serveur.

Voici la commande à utiliser :

```
service iptables save
```

Faites un essai, et redémarrer votre serveur afin de confirmer que les règles d'accès sont toujours actives.

Commandes utiles

Liste des règles :

```
iptables -L
```

Remise à zéro des règles :

```
iptables -F
```