Salt

Salt est un logiciel de gestion de configuration comme Puppet ou Ansible.

Je l'utilise chez Framasoft et sur mon infra personnelle parce que je l'aime bien :

- rapide ;
- très bien documenté ;
- syntaxe claire, accessible mais néanmoins flexible et puissante.

Changer ses mots de passe en masse

Je change mes mots de passe régulièrement (une fois par an environ). C'est toujours galère à faire quand on gère une tripotée de serveurs (entre les serveurs physiques et les VMs, on en est à plus de 100 serveurs chez Framasoft).

Avant, je faisais ça à la main : je lançais mssh sur 4, 6 ou 8 serveurs à la fois, et je modifiais mon mot de passe à la main. Mais ça, c'était avant.

Salt à la rescousse

Pour changer le mot de passe de l'utilisateur `bar` sur le serveur `foo` avec salt, on fait :

```
salt foo shadow.set_password bar "$6$selselse$HASHEDPASSWORD"
```

`$6$selselse$HASHEDPASSWORD` correspond à votre mot de passe salé et hashé. Vous retrouvez un brol du genre dans votre `/etc/shadow` (Oh ! Vous avez remarqué ? C'est le nom du module salt qui permet de modifier votre mot de passe ! C'est bien fait quand même □□)

Pour créer l'ensemble `$6$selselse$HASHEDPASSWORD`, vous pouvez utiliser python (nécessite le module passlib, fourni par le paquet Debian `python3-passlib`) :

```
python3 -c "from passlib.hash import sha512_crypt; print(sha512_crypt.hash('LE_PASSWORD',
rounds=5000))"
```

Bon, on sait comment faire, mais on ne va pas s'amuser à taper 100 fois ces commandes !

Salt permet de vérifier que les minions (les agents Salt) répondent bien avec cette commande :

```
salt foo test.ping
```

Ce qui donne :

```
foo:
  True
```

On va changer le format de sortie :

```
salt foo --out text test.ping
```

Ce qui nous donne :

```
foo: True
```

Bien ! On peut pinguer d'un coup tous les minions avec :

```
salt \* --out text test.ping
```

On a donc la liste des minions, la commande pour changer le mot de passe... on va mixer tout ça :

```
salt \* --out text test.ping | \
  sed -e "s@\([^:]*\):.*@echo salt \1 shadow.set_password bar \\\\\"\$(python3 -c \"from
passlib.hash import sha512_crypt; print(sha512_crypt.hash('LE_PASSWORD',
rounds=5000))\")\\\\\"@"
```

1ère regex : on dégage les `: True`, et la deuxième, on enrobe le nom du minion pour que ça nous donne un truc comme :

```
echo salt foo shadow.set_password bar \"$(python3 -c "from passlib.hash import sha512_crypt;
print(sha512_crypt.hash('LE_PASSWORD', rounds=5000))")\"
```

Quand on exécute ça, ça donne un truc genre :

```
salt foo shadow.set_password bar
"$6$8qJhzAi6$.O8bOisJaM9fH05aXx7xnKXOVFoI9CRzjORFWDqoPR/TBOiYVZUEJKtUKirNMyaZJvJMYPVUMhnNry9QP
JgHK/"
```

Bien évidemment, on va mettre ça dans un fichier qu'on va éditer pour modifier le mot de passe (bah oui, on va quand même pas mettre le même mot de passe sur tous les serveurs).

```
salt \* --out text test.ping | \
  sed -e "s@\([^:]*\):.*@echo salt \1 shadow.set_password bar \\\\\"\$(python3 -c \"from
passlib.hash import sha512_crypt; print(sha512_crypt.hash('LE_PASSWORD',
```

```
rounds=5000))\")\\\\\"@" > /tmp/chpasswd.txt
```

On édite `/tmp/chpasswd.txt` pour mettre ses mots de passe bien comme il faut puis :

```
bash /tmp/chpasswd.txt | bash
```

Le `echo` va nous sortir la commande kivabien qui sera interprétée par bash. Le bout de python transformera le mot de passe en hash dans le format kivabien pour le fichier `/etc/shadow` et la commande salt sera lancée sur chaque minion.

Révision #2
Créé 2023-01-19 09:59:19 CET par Luc
Mis à jour 2024-10-30 12:09:38 CET par Luc