

Comment automatiser les commandes scriptées sur un Cisco ASA via ssh

Table of Contents

[Le contenu de la crontab root](#)

[Le contenu de /home/jpavlov/expectScript.sh](#)

[Le contenu de /home/jpavlov/commandScript.sh](#)

L'objectif est assez simple, envoyer 4 ou 5 commandes à un Cisco ASA à partir d'un CentOS dans une tâche cron, cependant:

- L'ASA ne considère pas les points-virgules comme des substituts aux retours. Cela signifie qu'on ne peut pas utiliser une commande standard sur une seule ligne dans l'appel ssh. chaque commande doit être envoyée avec un retour réel.
- Bien que l'ASA puisse créer un ensemble de clés à utiliser avec ssh, il ne peut stocker aucune autre clé pour les hôtes distants, comme cela est nécessaire lors de l'utilisation de ssh basé sur des clés à partir d'une machine Linux distante. Donc, pas de ssh basé sur une clé ; il doit s'agir uniquement d'un mot de passe. Et il n'y a aucun moyen de fournir le mot de passe pour ssh à l'avance.

Pour cela, le programme **expect**, bien qu'il semble avoir été conçu à l'origine pour être utilisé avec des modems, convient parfaitement. Il agit comme un wrapper, presque comme un sous-shell, dans lequel on insère certaines instructions et commandes. Puisque le sous-shell virtuel a le contrôle, il exécute les commandes et écoute les invites apparaissant dans STDOUT et peut y répondre. C'est un moyen de capturer l'invite de mot de passe avec ssh !

Cependant le programme **expect** ne peut pas gérer seul le comportement de l'ASA. Puisque peu importe la façon dont on essaye de séparer les commandes que l'on souhaite passer à l'ASA, **Expect** ne cesse de les rassembler, séparées par des points-virgules (ce qui ne fonctionne pas avec l'ASA). La seule façon de contourner ce problème est donc de placer les commandes de session réelles dans un sous-script et d'appeler ce sous-script à partir d'expect. La logique ressemble à ceci :



crontab → expectScript.sh → commandScript.sh

Voici les mécanismes de tout cela:

Le contenu de la crontab root

```
6 */8 * * * /home/jpavlov/expectScript.sh
```

Le contenu de /home/jpavlov/expectScript.sh

```
#!/bin/bash
PASSWORD="asaSshPassword"
expect -d -c "set timeout -1;\
spawn /home/jpavlov/commandScript.sh;\
match_max 100000;\
expect *password:*\
send -- $PASSWORD\r;\
interact;"
```

Le contenu de /home/jpavlov/commandScript.sh

```
ssh routerUser@192.168.1.1 << EOF
en
routerEnablePassword
clear crypto ipsec sa peer 192.168.66.6
exit
exit
EOF
```



Il faut être prudent avec le filtre à l'intérieur du jeu de commandes **expect**. Le filtre prend en charge les expressions régulières et est sensible à la casse. Il faut s'assurer que le filtre corresponde, sinon cela ne fonctionnera pas, évidemment.



Il faut également s'assurer de faire un `chmod 700` sur les deux scripts pour le garder à l'abri des regards indiscrets. Il est déconseillé de faire cela sur des serveurs multi-usages, mais c'est moins préoccupant sur une box restreinte.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=howto:asa-shell-script>

Last update: **2025/02/19 10:59**

