[Cisco] IOS

Introduction

IOS pour **Internetwork OS** est le système d'exploitation de Cisco présent sur ses équipements physique et sur Packet Tracer évidemment.



Lignes et interfaces

Sur les équipements, on retrouve deux types de ports :

- Les **interfaces** qui sont les ports réseaux où seront connectés les utilisateurs finaux ou les switch qui sont *in-band*.

- Les **lignes** qui sont soit physiques (port USB ou série) ou virtuelles (*VTY*) qui permettent de se connecter à la **console** (aussi appelé *rollover*).

Les contextes

Il existe dans IOS trois contextes permettant l'exécution de certaines commandes.

Mode racine

C'est le mode permettant l'exécution de toutes les commandes relatives à l'équipement lui-même.

Il est représenté par le prompt suivant :

```
Hostname#
```

Mode de configuration global

C'est le mode permettant la configuration relative à l'équipement lui-même.

Il est représenté par le prompt suivant :

```
Hostname(config)#
```

Pour l'activer il faut exécuter la commande suivante :

```
configure terminal
```

Mode de configuration spécifique

C'est le mode permettant la configuration d'une **ligne** ou d'une **interface** spécifique.

Il est représenté par le prompt suivant où le **X** est le composant spécifique sélectionné :

```
Hostname(config-X)#
```

Pour accéder à une configuration spécifique d'interface on utilise la commande suivante :

```
interface <INTERFACE_NAME> <INTERFACE_NUMBER>
```

Pour accéder à une configuration spécifique dune ligne on utilise la commande suivante :

```
line [console/vty] [NUMBER=0]
```

Niveaux de privilège d'exécution

Il existe dans IOS deux niveaux de privilèges d'exécution : le mode d'exécution **utilisateur** et **privilégié**.

Mode d'exécution utilisateur

Ce mode est celui sélectionné par défaut si un mot de passe a été définit pour le mode d'exécution privilégié.

On dit qu'il a un niveau de privilège à **1**.

Il permet d'accéder en **lecture-seule** à la configuration de l'appareil et permet aussi d'exécuter des commandes de **diagnostic**.

Il est représenté par le caractère **>** dans le prompt :

```
Hostname(config)>
```

Mode d'exécution privilégié

Ce mode permet l'**exécution de l'ensemble des commandes** (dans le contexte approprié).

On dit qu'il a un niveau de privilège à **15**.

Pour y accéder, il faut exécuter la commande suivante :

```
enable
```

Un **mot de passe** peut être définit pour accéder à ce mode grâce à la commande suivante :

```
secret <YOUR_PASSWD>
```

> La commande secret a succédée à la commande password qui n'était pas sécurisée et permettait de voir les mots de passe en clair dans la config.

Le mode d'exécution privilégié est représenté par le caractère **#** dans le prompt :

```
Hostname(config)#
```

VTY

Il existe une autre manière de se connecter à la console plutôt que de passer par le **rollover** qui est *out-of-band* (qui est un port d'administration).

Cette méthode c'est le **VTY**, qui permet de se connecter à la console en passant par une interface.

Le **VTY** permet entre autre, de pouvoir se connecter à distance à la console.

Cela est possible car l'équipement est capable de détecter qu'une trame lui est destinée (en comparant l'adresse MAC de destination à la sienne).

Par défaut il y a 5 **VTY** ouverts sur un équipement nommés **VTY0** puis **VTY1** jusqu'à 4.

Les configurations

Il existe deux types de configuration sur IOS : la **running-config** et la **startup-config**

La running-config

Il s'agit de la configuration actuelle du système qui est mis à jour à chaque exécution de commande sur le système.

Elle est stockée dans la **RAM** et peut, voire doit être sauvegardée car un redémarrage la supprime complètement.

La startup-config

Il s'agit de la configuration de la version du dernier redémarrage qui n'est mis à jour que lors d'un redémarrage contrôlé ou de l'import manuel.

Elle est stockée dans la **NVRAM** (Non-Volatile RAM) et est donc persistante après le redémarrage.

Sauvegarder la running-config dans la startup-config

```
copy running-config startup-config
```

Il existe une commande alternative bien plus courte :

```
wr
```

Aide à la saisie

En cas de difficulté sur les commandes disponibles dans un contexte ou sur un commande spécifique, le caractère **?** peut nous aider en nous donnant des informations.

Il peut être utilisé tout seul :

```
?
```

Ou dans une commande de cette façon :

```
show ip ad?
```

Commandes utiles

Afficher les interfaces

Permet d'afficher des informations sur les interfaces de l'équipement comme l'adresse IP ou son status :

```
show ip interface brief
```

Afficher la table MAC

Cette commande ne fonctionnera que sur un switch :

```
show mac address-table
```

Afficher les ports ouverts et services actifs

Vous pouvez afficher les ports actuellement utilisés par vos services actifs avec cette commande :

```
show ip ports all
```

Do

Permet d'exécuter une commande de contexte racine dans un autre contexte :

```
do <COMMAND>
```

Hostname

Permet de changer le **nom d'hôte** de l'équipement :

```
hostname <HOSTNAME>
```

IP Domain

Permet de changer le **nom d'hôte sur le réseau** de l'équipement :

```
ip domain name <FQDN>
```

> Sur un équipement local, vous pouvez utilisez le hostname.local par exemple.

Motd

Permet de changer la bannière lors de la connexion à l'équipement :

```
banner motd <END_CHARACTER>
```

Exit

Permet de sortir du contexte actuel pour revenir au contexte précédent :

```
exit
```

End

Permet de revenir au contexte racine :

```
end
```

Show version

Permet d'afficher la version de l'image (OS) :

```
show version
```

Le pipe

Il permet de filtrer la sortie d'une commande (comme sous Linux) et admet différentes options.

Il a pour caractère **|** et bénéficie de plusieurs fonctionnalités.

Voici son usage :

```
<COMMAND> | <OPTION> <ARGUMENT>
```

Include

Permet d'agir comme la commande grep sous Linux dans son usage initial, c'est à dire pour capturer uniquement les lignes où le motif spécifié est présent :

```
<COMMAND> | include <MOTIF>
```

Exclude

Permet de faire l'inverse d'include, c'est à dire d'afficher toutes les lignes dont le motif spécifié n'est pas présent :

```
<COMMAND> | exclude <MOTIF>
```

Begin

Agit comme l'include mais affiche toutes lignes à partir de la ligne où le motif a été trouvée :

```
<COMMAND> | begin <MOTIF>
```

Section

Agit comme le begin mais n'affiche que la section du motif, c'est à dire s'il détecte des espaces ou des tabulations :

```
<COMMAND> | section <MOTIF>
```

Revision #22
Created 2023-09-27 19:16:24 UTC by Elieroc
Updated 2023-11-17 14:10:11 UTC by Elieroc