Commandes d’administration pour les routeurs et les commutateurs

Les VLANs

- Créer des vlans, sur le switch :

```
Switch# vlan database
Switch(vlan)# vlan 2 name compta
Switch(vlan)# vlan 3
```

Ici le vlan 2, nommé “compta” vient d’être créé, le nom est facultatif, le vlan 3 est aussi créé mais n’a pas de nom. NB le vlan 1 existe par défaut, il ne peut être supprimé ou renommé, toutes les interfaces sont par défaut attribuées au vlan 1.

- Attribuer des interfaces (fast ethernet) à des vlans, sur le switch :

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

Ici, on vient d'attribuer à l’interface fastEthernet 0/1 le vlan 2.

- Créer un port trunk pour faire passer plusieurs vlans sur un même port, sur le switch :

```
Switch(config)#interface fastEthernet 0/24
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport allowed vlan 1,2,3
```

Ici, on vient d’autoriser l’interface fa 0/24 à faire passer les vlans 1,2,3. On aurait pu écrire 1-3 au lieu de 1,2,3.

- Faire de routage inter vlan, sur le routeur :

Il faut relier l'interface configurée en mode trunk du switch à une interface du routeur ici fa 0/1.

```
Router(config)#interface fastEthernet 0/1
Router(config-if)#no shutdown
Router(config-if)#interface fastEthernet 0/1.1
Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip address 192.168.1.254 255.255.255.0
Router(config-if)#interface fastEthernet 0/1.2
Router(config-subif)#encapsulation dot1Q 3
Router(config-subif)#ip address 192.168.2.254 255.255.255.0
```

Ici, on vient de configurer le routage pour le vlan 2, (encapsulation dot1Q 2), la passerelle par défaut pour le vlan 2 est donc 192.168.1.254. De même pour le vlan 3, avec 192.168.2.254 comme passerelle par défaut. Et ainsi de suite pour les autres vlans. Il faut bien sûr que les vlans soient autorisés à utiliser le port trunk du switch relié au routeur.

Protection des accès

Mode console

Commandes à taper pour demander un mot de passe lors d'une connexion au port console.

```
Router(config)#line console 0
Router(config-line)#password cisco
Router(config-line)#login
```

Avec ces 3 lignes commandes, le mot de passe "cisco" nous sera demandé lors d'une connexion par le port console.

Mode Telnet (sessions distantes)

Commandes à taper pour demander un mot de passe lors d'une connexion à distance par telnet.

```
Router(config)#line vty 0 4
Router(config-line)#password distance
Router(config-line)#login
```

Avec ces 3 lignes commandes, le mot de passe "distance" nous sera demandé lors d'une connexion à distance par telnet. Il faut qu'un mot de passe pour passer en mode privilégié soit défini, sinon on restera bloqué en mode utilisateur.

Mode privilégié

Commandes à taper pour demander un mot de passe lors du passage du mode utilisateur au mode privilégié, en mode console ou en telnet.

```
Router(config)#enable password privilege
```

On vient de définir le mot de passe "privilege" qui nous sera demander suite à la commande (`router> enable`).

Mode local

On peut définir un ou plusieurs comptes locaux via la commande :

```
Router(config)#username yanis password azerty
```

On vient de définir le compte qui a pour identifiant “yanis” et mot de passe “azerty”. Il faut alors taper ces commandes pour choisir l'identification local (login + mot de passe) à la place de l’identification par mot de passe classique.

```
Router(config)#line vty 0 4
Router(config-line)#login local
```

Cela marche également pour le port console.

Chiffrer les mots de passe

Pour chiffrer les mots de passe afin qu’il n’apparaissent plus en clair dans le fichier de configuration, il faut utiliser cette commande :

```
Router(config)#service password-encryption
```

Cette commande chiffrera tous les mots de passe déjà créés, ainsi que ceux créés après. Si l’on veut créer un mot de passe enregistré en clair, il faudra taper la commande suivante puis créer ou recréer le mot de passe.

```
Router(config)#no service password-encryption
```

Récupération

Pour un routeur

Si l'on a oublié le(s) mot(s) de passe d'un routeur et donc que l'on a plus accès au routeur, il faut le redémarrer et appuyer sur les touches " Ctrl " + " Pause " durant les 60 secondes suivant son démarrage. Cela nous permet d'aller dans le mode rommon. Dans ce mode, il faut taper les commandes suivantes afin de forcer le routeur à démarrer sur une séquence vierge.

```
rommon 1 > confreg 0x2142
rommon 1 > boot
```

La première ligne va forcer le routeur à booter en ignorant la NVRAM, une fois le routeur démarrer il faut penser à paramétrer le routeur pour qu'au prochain redémarrage il boot sur la NVRAM.

```
Router(config)#config-register 0x2102
```

Une fois l'accès au routeur repris, on peut importer l'ancienne version afin d'écraser les anciens mots de passe. Pour ce faire :

```
Router#copy startup-config running-config
```

Puis sauvegarder la nouvelle version avec les mots de passe modifier avec :

```
Router#copy running-config startup-config
```

Pour un commutateur

Une fenêtre hyperterminal doit être activée sur l'ordinateur connecté au commutateur par le port console. Le commutateur doit être mis hors tension en débranchant le câble d'alimentation puis tout en maintenant enfoncé le bouton «MODE» situé sur la face avant, il faut mettre le commutateur sous tension. Relâchez le bouton «MODE» dès que la LED « STAT » change d'état. Il faut ensuite taper :

```
> flash_init
> load_helper
```

```
> rename flash:config.text flash:config.old
> boot
```

Le commutateur va donc démarrer en ignorant le fichier de config avec les mots de passe. Une fois démarré, on peut récupérer le fichier de configuration pour écraser les anciens mots de passe afin de ne pas perdre toute la configuration.

```
Switch# copy flash:config.old system:running-config
// “Modifier les mots de passe pour ne plus être bloqué”
Switch# copy running-config startup-config
Switch# delete flash:config.old
```

On sauvegarde donc la version propre de notre routeur et l’on détruit l’ancien fichier de configuration, maintenant plus utile.

Les ACL

Les règles sont lues les unes après les autres, selon l'ordre avec lequel elles ont été tapées, dès qu'une règle est satisfaite, la règle est appliquée (autoriser ou bloquer). Il est donc important de taper les règles les plus précises en premier jusqu'aux plus générales en dernier. Règle implicite en fin d'ACL : deny any. Pour les ACL numériques, il est impossible de modifier les règles ou l'ordre des règles. Il faut donc écrire les règles à l'avance dans un fichier texte, puis les injecter dans le routeur. Pour supprimer l' ACL n°5 il faut écrire :

```
Router(config)#no access-list 5
```

Les ACL standards numériques

Listes de contrôle d'accès avec des paramètres simples, permet d'autoriser ou de bloquer des réseaux grâce à leur adresse IP. Numéro de l'ACL entre 1 et 99 inclus.

```
Router(config)#access-list n°ACL(1-99) {deny | permit | remark} @IPsource, Masque générique
```

Exemple :

```
Router(config)#access-list 1 deny 192.168.0.0 0.0.255.255
```

On vient ici d'interdire, tous paquets dont l'adresse IP source commence par 192.168.
Il faudra penser à ajouter un "`permit any`" à la fin, sinon tous les réseaux seront bloqués.

Appliquer des ACL aux interfaces

Un fois les ACL créées, il faut les appliquer à des interfaces pour qu'elles soient appliquées.

```
Router(config)#interface fa 0/1
Router(config-if)#ip access-group 1 in
```

On vient d'appliquer l'ACL numéro 1, précédemment créée, sur l'interface fa 0/1, pour le trafic entrant. Pour le trafic sortant, il faut remplacer "in" par "out".

Le NAT

Le nat permet de changer les adresses IP source et/ou destination des paquets. Cela de manière statique ou dynamique.

IP masquerading

Permet de traduire un grand nombre d'adresses IP source, souvent privées (RFC 1918), en une seule adresse IP source routable sur internet. Pour ce faire, il faut définir un pool d'adresse IP, dans ce cas composé d'une seule adresse IP.

```
router(config)# ip nat pool RFC1918 87.77.77.77 87.77.77.77 netmask 255.255.255.0
```

RFC1918 est le nom du pool, 87.77.77.77 est l'adresse IP qui va être routée sur internet. Puis, il faut définir, grâce à une ACL, les adresses IP qui pourront être traduites.

```
router(config)# ip nat inside source list 1 pool RFC1918 overload
```

Le terme overload permet d'associer plusieurs adresses locales interne à une adresse globale interne. Ici, plusieurs hôtes pourront prendre l'adresse IP 87.77.77.77.

L’IPv6

DHCPv6 avec état

Le DHCPv6 avec état fournit tous les paramètres IP.

Pour ce faire :

```
router(config)# ipv6 dhcp pool statefull
router(config-dhcpv6)# dns-server 2001:4860:4860::8888
router(config-dhcpv6)# prefix-delegation pool state
router(config-dhcpv6)# exit
router(config)# ipv6 local pool state 2001:db8:0:10::/64 64
router(config)# int fa 0/0
router(config-int)# ipv6 dhcp server statefull
router(config-int)# ipv6 nd managed-config-flag
```

Ici, on crée 2 pools, le premier pool est un pool dhcp nommé “statefull” qui est à renseigner sur l’interface. Et le second pool state qui définit le préfix IPv6.