Mini réseau d'entreprise

MIF21 - réseaux par la pratique

Univ. Claude Bernard Lyon 1

2014-2015

Fabien Rico fabien.rico@univ-lyon1.fr
Jacques Bonneville jacques.bonneville@univ-lyon1.fr
Olivier Glück Olivier.Gluck@univ-lyon1.fr

Objectifs du cours

- Configuration DHCP
- Configuration ACL
- Configuration NAT
- Configuration Tunnel

Dynamic Host Configuration Protocol

- Permet d'attribuer une adresse aux machines.
- Ces adresses sont valides pendant un temps fini (baux DHCP).
- Plusieurs configurations sont attribuées en même temps :
  - proxy,
  - route par défaut,
  - serveur ntp,
  - fichier de démarage BOOTP,
  - ...

Configuration

- Activation du service

```
Router(config)# service dhcp
```

- Configuration d'un pool d'adresses

```
Router(config)# ip dhcp pool nom
Router(dhcp-config)# network adresseRes [mask|/longueur]
Router(dhcp-config)# default-router adresse1 [address2 ...]
Router(dhcp-config)# dns-server adresse1 [adresse2 ...]
Router(dhcp-config)# domain-name domaine
Router(dhcp-config)# netbios-name-server adresse1 [adresse2 ...]
```

- On choisit une durée de validité

```
Router(dhcp-config)# lease (jours [heures] [minutes] | infinite)
```

- Certaines adresses du pool sont retirées car réservées à d'autre appareils (serveur, routeur, ...)

```
Router(config)# ip dhcp excluded-address adresseDebut [adresseFin]
```

Liste de control d'acces

Les ACLs servent tout d'abord à attribuer des droits à des groupes. Dans le cadre du réseau, cela permet

- de reconnaitre des paquets : IP source, port source, IP destination, port destination ;
- d'autoriser ou pas le passage ;
- d'appliquer un traitement à certains paquets.

Il y a plusieurs type d'ACL :

- standard qui s'appliquent à la source uniquement ;
- étendue qui s'appliquent aux 4 valeurs ;
- dynamique (refléxive, CBAC, ZBF).

ACL

C'est une liste,

- lue dans un certain ordre (voir le numéro de séquence) ;
- chaque item va correspondre à des adresses et l'accepter (permit) ou le refuser (deny) ;
- à la fin, il y a une règle implicite qui refuse tout.

En IPV4, il y a 2 façons de les écrire :

- `access-list ...` qui permet de créer un item à la fois,
- `ip access-list ...` qui rentre dans un module de configuration de la liste entière.

ACL standard

Numéro entre 1 et 99

```
Router(config)# access-list acl-num (permit|deny)
                            source [source-wildcard] [log]
```

Ou

```
Router(config)# ip access-list standard (NAME|acl-num)
Router(config-std-nacl)# sequence-num (permit|deny)
                            source [source-wildcard] [log]
Router(config-std-nacl)# 10 permit 10.15.0.0 0.0.255.255
```

Vérification des ACLs

```
Router# show access-lists [ACL-num | NAME]
```

ACL étendue

Numéro entre 100 et 199

```
Router(config)# access-list ACL-num (permit|deny) protocol source src-wildcard
            destination dest-wildcard (eq|neq|gt|lt|range) port-num [established]
```

Ou

```
Router(config)# ip access-list extended (NAME|ACL-num)
Router(config-ext-nacl)# sequence-num (permit|deny) protocol source src-wildcard
             destination dest-wildcard (eq|neq|gt|lt|range) port-num [established]
Router(config-ext-nacl)# access-list 110 permit tcp 10.15.0.0 0.0.255.255 eq www
                                                    any gt 1024
```

Vérification

```
Router# show access-lists [ACL-num | NAME]
```

Application d'une ACL

- Sur les interfaces en entrée ou en sortie

```
Router(config-if)# ip access-group
(ACLname|ACLnum) (in|out)
```

- Sur les accès telnet

```
Router(config)# line vty 0 4
Router(config-line)# access-class ACLnum (in|out)
```

ACL a état

Les ACLs vues ne filtrent qu'à partir de la source, de la destination et des ports utilisés. Mais cela ne permet pas de tout faire, par exemple : il est impossible de ne rien laisser entrer sauf les réponses à des connexions depuis l'intérieur.
Pour cela il faut des **ACLs à état** : il faut que le pare-feux se souvienne de ce qui est sorti pour en autoriser le retour. C'est ce que permet le CBAC (context based access control) :

- On crée une *inspection*

```
Router(config)# ip inspect name nom
(tcp|udp|icmp|...) [timeout tempsvalidite]
```

- On l'applique sur une interface (en général celle de sortie du paquet)

```
Router(config-if)# ip inspect nom (in|out)
```

Ce système modifie les ACLs étendues pour autoriser les paquets de retour pendant un temps limité.

Traduction d'adresse statique

Traduction entre 2 adresses, une publique et une privée. Par exemple pour un serveur.

- Déclaration des adresses à traduire

```
Router(config)# ip nat inside source static
                                    ipPrivée ipPublic
```

- Même chose mais pour un port uniquement

```
Router(config)# ip nat inside source static (tcp|udp)
                                ipPriv portpriv ipPub portpub
```

- Il faut déclarer les interfaces interne et externe

```
Router(config)# interface type number
Router(config-if)# ip nat inside
Router(config)# interface type number
Router(config-if)# ip nat outside
```

NAT dynamique, PAT

- Création d'un pool d'addresses publiques utilisables

```
Router(config)# ip nat pool NAME start-ip end-ip (netmask netmask
                                  |prefix-length prefix-length)
```

- Une ACL pour reconnaitre les adresses privées à traduire

```
Router(config)# access-list ACLnum permit source [src-wildcard]
```

- Déclarer la traduction

```
Router(config)# ip nat inside source list ACLnum (interface interface
                                        | pool nomDuPool) [overload]
```

- Il faut déclarer les interfaces de sortie et d'entrée

```
Router(config)# interface type number
Router(config-if)# ip nat inside
Router(config)# interface type number
Router(config-if)# ip nat outside
```

Tunnel

C'est un moyen d'encapsuler un paquet dans un autre. On peut ainsi créer 2 interfaces virtuelles simulant une liaison directe entre 2 routeurs qui passe en réalité par internet ou par une liaison louée.

- On peut transporter des paquets ipv6 à travers un réseau ipv4, ou créer une liaison entre 2 parties d'un réseau utilisant des adresses privées.
- C'est utilisé par beaucoup de VPN.

Tunnel gre

Protocol CISCO, il permet d'encapsuler plusieurs autres protocols :

- *gre ip* pour encapsuler des paquets IP,
- *ipv6ip* pour encapsuler les paquets IPv6 dans des paquets IPv4.

Utilisation

Sur les 2 routeurs, on crée une interface virtuelle

```
R1(config)# interface tunnel number
R1(config-if)# tunnel mode gre ip
R1(config-if)# ! adresse public du routeur
R1(config-if)# tunnel source 134.214.12.1
R1(config-if)# ! adresse public du l'autre
R1(config-if)# tunnel destination
129.14.1.1
R1(config-if)# ! adresse privée de
l'interface
R1(config-if)# ip address 10.1.1.1
255.255.255.0
```

```
R2(config)# interface tunnel number
R2(config-if)# tunnel mode gre ip
R2(config-if)# ! adresse public du routeur
R2(config-if)# tunnel source 129.14.1.1
R2(config-if)# ! adresse public du l'autre
R2(config-if)# tunnel destination
134.214.12.1
R2(config-if)# ! adresse privée de
l'interface
R2(config-if)# ip address 10.1.1.2
255.255.255.0
```

Puis on modifie les routes sur les routeurs :

- En considérant que 10.25.0.0/16 est du coté de R2

```
R1(config)# ip route add 10.25.0.0 255.255.0.0
10.1.1.2
```

- Et que 10.15.0.0/16 est du coté de R1

```
R2(config)# ip route add 10.15.0.0 255.255.0.0
10.1.1.1
```

Détail des commandes

- Pour choisir les paquets transportés

```
Routeur(config-if)#tunnel mode (gre ip|ipv6ip|...)
```

- Pour choisir l'adresse de source

```
Routeur(config-if)# tunnel source (@ip | @ipv6 |
int-type number)
```