COURS 4
ACL - Liste de Contrôle d'Accès

1 – Qu'est-ce qu'une ACL ?

Une liste de contrôle d'accès (ACL - Access Control List) est un ensemble de règles utilisé dans les réseaux informatiques pour contrôler le flux de trafic entrant ou sortant d'un routeur ou d'un commutateur en fonction de critères tels que :

- Les adresses IP sources et de destination,
- Les ports de protocole,
- Les types de trafic, …

Attention

Une mauvaise configuration des ACL pourrait entraîner des problèmes de connectivité ou de sécurité.

2 – Dans quel cas utilise-ton des ACL ?

Sécurité réseau

Vous pouvez configurer une ACL pour bloquer le trafic provenant d'adresses IP spécifiques ou pour autoriser uniquement certaines applications à accéder à certaines ressources.

Qualité de service (QoS)

Les ACL sont utilisées pour prioriser certains types de trafic par rapport à d'autres, garantissant ainsi un niveau de performance optimal pour les applications sensibles à la latence, comme la voix sur IP (VoIP) ou la vidéo en streaming.

Filtrage de contenu

Les ACL peuvent être utilisées pour bloquer l'accès à certains contenus en ligne en fonction de mots-clés, de domaines ou d'adresses IP.

3 – Les ACL standards

ACL standard : - uniquement en IPv4 (filtrage au niveau 3 du modèle OSI)
- examine seulement l'adresse IP source car le filtrage se fait sur la couche 3 (OSI).

Router(config)#**access-list** *numéro* {**deny**|**permit**} *@IP_source wildcard* [**log**]

Numéro (seulement en IPv4) : Numéro de la liste ACL compris entre 1et 99 ou entre 1300 et 1999.

@IP source : Identifie le réseau source ou l'adresse hôte à filtrer. On peut utiliser **any** pour spécifier tous les réseaux ou **host** + **@IP** pour identifier une adresse IP spécifique.

wildcard : Masque générique pour l'adresse IP source (0.0.0.0 signifie que tous les bits doivent correspondre).

log (optionnel) : Permet de consigner les correspondances.

Exemple

```
Router(config)#access-list 1 deny host 192.168.1.100
Router(config)#access-list 1 permit any
```

Interprétation

L'ACL standard 1 empêche seulement le trafic venant de l'hôte 192.168.1.100. Toutes les autres adresses peuvent passer.

Attention, si on change l'ordre, cela ne donne pas le même résultat.

```
Router(config)#access-list 1 permit any
Router(config)#access-list 1 deny host 192.168.1.100
```

En effet, la première règle autorise toutes les adresses, elle sera toujours validée en premier donc la seconde règle ne sera jamais interprétée.

L'ordre des règles en très important !

Dès qu'une règle est validée on sort de la liste et on n'exécute pas les règles suivantes.

Important : Une liste ACL doit avoir au moins une déclaration d'autorisation sinon tout le trafic sera refusé en raison de l'instruction de refus implicite. Par défaut, cette déclaration est automatiquement implicite à la fin d'une ACL même si elle est masquée et non affichée dans la configuration.

Exercices sur les ACL standards

Proposez les ACL standards pour les cas suivants :

Cas 1

ACL (numéro 5) : autorisez seulement l'adresse IP source 192.168.1.3.

```
Router(config)#access-list 5 permit host 192.168.1.3
```

Cas 2

ACL (numéro 10) : bloquez la plage d'adresses IP sources 192.168.1.0 /24 (voir chapitre 6 – Masque générique)

```
Router(config)#access-list 10 deny 192.168.1.0 0.0.0.255
```

Cas 3

ACL (numéro 15) : autorisez tout le trafic provenant du sous-réseau 10.0.0.0 /8

```
Router(config)# access-list 15 permit 10.0.0.0 0.255.255.255
```

Cas 4

ACL (numéro 20) : bloquez une seule adresse IP (192.168.1.4)

```
Router(config)# access-list 20 deny host 192.168.1.4
```

Cas 5

ACL (numéro 25) : autorisez un groupe d'adresses spécifique de 192.168.1.1 à 192.168.1.15

```
Router(config)# access-list 25 permit 192.168.1.0 0.0.0.15
```

4 – Exercices sur les ACL étendues

ACL étendue : IPv4 et IPv6 (filtrage au niveau 3 et supérieures du modèle OSI)

Router(config)#**access-list** *numéro* {**deny ou permit**} *protocole @IP_source wildcard* [*opérateur port*] *@IP_destination wildcard* [*opérateur port*] [**log**]

Numéro (seulement en IPv4) : Numéro de la liste ACL compris entre 100 et 199 ou entre 2000 et 2699.

permit ou **deny** : Spécifie si le trafic correspondant doit être autorisé ou bloqué.

protocole : (couche 2, 3 ou 4 du modèle OSI) icmp, ip, tcp, udp, eigrp, ospf, … ou un nombre représentant le protocole utilisé.

@IP source : Identifie le réseau source ou l'adresse hôte à filtrer. On peut utiliser **any** pour spécifier tous les réseaux ou **host** + **@IP** pour identifier une adresse IP spécifique.

wildcard : Masque générique pour l'adresse IP source (0.0.0.0 signifie que tous les bits doivent correspondre).

opérateur : lt (less than - moins que), gt (greater than - plus que), eq (equal - égal à), neq (not equal - non équivalent), range (inclusive range - étendue d'une plage).

port : valeur nominative ou numéraire : de 0 à 65535 ou, par exemple, www, telnet, ftp, etc.

@IP_destination : Adresse IP destination du trafic.

log (optionnel) : Permet de consigner les correspondances.

Exemple

```
Router(config)#access-list 102 deny tcp any any eq 23
Router(config)#access-list 102 permit ip any any
```

Interprétation

Pour la liste ACL 102 étendue (numéro 102 compris entre 100 et 199), le trafic TCP provenant de n'importe quelle adresse IP source et à destination de n'importe quelle adresse IP de destination, mais ayant le port de destination égal à 23 (le port Telnet), sera bloqué.

deny tcp : Vous refusez le trafic TCP.

any any : Vous ciblez n'importe quelle @IP source et n'importe quelle @IP de destination.

eq 23 : "eq" signifie "equal" (égal), et 23 est le numéro de port pour le service Telnet. Donc, cela indique que le trafic doit avoir comme port de destination le port 23, qui est associé au service Telnet.

Exercices sur les ACL étendues

Proposez les ACL étendues pour les cas suivants :

Cas 1

ACL (numéro 100) : autorisez le trafic TCP sortant avec le port de destination 80 (HTTP), permettant ainsi aux utilisateurs de naviguer sur le Web.

```
Router(config)#access-list 100 permit tcp any any eq 80
```

Cas 2

ACL (numéro 110) : bloquez tout le trafic ICMP entrant, ce qui peut contribuer à réduire les attaques de déni de service (DOS).

```
Router(config)#access-list 110 deny icmp any any
```

Cas 3

ACL (numéro 120) : autorisez le trafic entrant pour un tunnel VPN avec les protocoles ESP (Encapsulating Security Payload) et UDP (port 500) vers l'adresse IP de destination 200.0.153.20.

```
Router(config)#access-list 120 permit esp any host 200.0.153.20
Router(config)#access-list 120 permit udp any host 200.0.153.20 eq 500
```

Cas 4

ACL (numéro 140) : autorisez le trafic UDP sortant avec le port de destination 53 (DNS), permettant aux utilisateurs de résoudre des noms de domaine en adresses IP.

```
Router(config)#access-list 140 permit udp any any eq 53
```

Exercices (un peu plus) complexes sur les ACL étendues

Cas 1

ACL (numéro 150) : autorisez le trafic HTTP sortant (port 80) tout en bloquant le trafic Telnet entrant (port 23).

```
Router(config)#access-list 150 permit tcp any any eq 80
Router(config)#access-list 150 deny tcp any any eq 23
```

Cas 2

ACL (numéro 160) : autorisez le trafic ESP pour le VPN vers l'adresse IP de destination 194.0.5.105, bloquez le trafic FTP (port 21) et autorisez le trafic DNS (port 53).

```
Router(config)#access-list 160 permit esp any host 194.0.5.105
Router(config)#access-list 160 deny tcp any any eq 21
Router(config)#access-list 160 pemit udp any any eq 53
(Router(config)#access-list 160 pemit tcp any any eq 53)        --> rares cas où tcp est utilisé avec DNS
```

Cas 3

ACL (numéro 170) : autorisez uniquement le trafic HTTP (port 80) entrant vers le serveur Web interne (192.168.1.10), tandis que le reste du trafic vers ce serveur web est bloqué.

```
Router(config)#access-list 170 permit tcp any host 192.168.1.2 eq 80
Router(config)#access-list 170 deny ip any host 192.168.1.2
```

Dans le cas 3, l'ordre des règles est important.

Cas 4

ACL (numéro 180) : bloquez le trafic HTTPS (port 443) ainsi que le trafic des ports 3470 à 3479, ciblant potentiellement les services de réseau social et de communication en temps réel.

```
Router(config)#access-list 180 deny tcp any any eq 443
Router(config)#access-list 180 deny tcp any any range 3470 3479
```

Cas 5

ACL (numéro 190) : autorisez le trafic UDP dans la plage de ports 10000 à 20000 ainsi que le trafic TCP et UDP sur le port 5060 (utilisé pour la VoIP). Tout autre trafic est bloqué, permettant de prioriser le trafic VoIP.

```
Router(config)#access-list 190 permit udp any any range 10000 20000
Router(config)#access-list 190 permit tcp any any eq 5060
Router(config)#access-list 190 permit udp any any eq 5060
Router(config)#access-list 190 deny ip any any
```

5 – Nommage d'une ACL

Si l'on ne souhaite pas numéroter les ACL on peut alors les nommées. En IPv4, il faudra alors préciser si on est dans le cas "**standard**" ou "**extended**". Il n'est pas nécessaire de préciser le cas en IPv6 puisqu'on utilise seulement des ACL étendues.

Exemple de configuration

Création d'une ACL nommée "DENY_PC1" qui bloque l'adresse 192.168.5.2. L'ACL est ajouté à l'interface F1/0 en sortie.

```
R1(config)#ip access-list standard DENY_PC1
R1(config-std-nacl)#deny host 192.168.5.2
R1(config-std-nacl)#exit
R1(config)#int f1/0
R1(config-if)#ip access-group DENY_PC1 out
```

Définition de la liste avec une ou plusieurs règles.

Configuration de l'interface.

C'est quoi ce "out" ?

Cela correspond à une direction de filtrage du trafic au niveau de l'interface.
Une liste d'accès s'applique sur une interface soit pour le trafic entrant "**in**", soit pour le trafic sortant "**out**".
Il faudra toujours le spécifier au niveau de l'interface par rapport au routeur.

La prise de décision

Le routeur parcourt la liste d'accès et valide chaque règle jusqu'à trouver une correspondance.

Si une correspondance est trouvée, le routeur prend la décision permit ou deny correspondante.

Il ne lit pas la suite de la liste.

Une ACL se termine toujours par une règle deny any implicite.

Important

Placez les ACL standard aussi près que possible de la destination.
Placez les ACL étendues aussi près de la source que possible.
Les paquets générés par le routeur ne sont pas filtrés par les listes de contrôle d'accès sortantes

6 – Masque générique = "wilcard mask" (masque de filtrage)

Il ne faut pas confondre un masque générique (wilcard mask) avec un masque de sous-réseau (subnet mask). Un masque générique est un masque de filtrage.

Le principe

Si un bit a la valeur 0 dans le masque => vérification de ce bit sur l'adresse IP de référence.
Si un bit a la valeur 1 dans le masque => Pas de vérification.

Exemple

Je souhaite bloquer la plage d'adresses IP sources 192.168.1.0 /24.
Cela suppose de bloquer toutes les adresses comprises entre 192.168.1.0 et 192.168.1.255

Dans ces adresses IP, il n'y a que le dernier octet qui varie (de 0 à 255) donc on prend toutes les valeurs de cet octet. => bit à 1 au niveau du dernier octet du masque générique.

Dans les adresses on constate une partie fixe "192.168.1", c'est elle que le routeur va essayer de rechercher

⇨ Il faut vérifier les trois premiers octets => Bit à 0 au niveau du masque générique

Masque générique : **0 . 0 . 0 . 255**

Méthodologie :

On dispose de 254 hôtes sur ce réseau => 254 hôtes = 2 ^8 – 2
=> 8 bits à 1 (on ne vérifie pas la partie "Host ID")
=> il reste 32 – 8 = 24 bits à 0
=> masque générique = 0.0.0.1111 1111
= 0.0.0.255

Exercice corrigé

Calcul du masque générique dans le cas d'un réseau avec 30 hôtes.

30 hôtes = 2 ^5 – 2 => 5 bits à 1 (on ne vérifie pas la partie "Host ID")
=> il reste 32 – 5 = 27 bits à 0
=> masque générique = 0.0.0.0001 1111
= 0.0.0.32