**Les ACL étendues**

Une *ACL* ***PEUT*** *servir à filtrer, mais elle peut aussi faire d’autres choses. De façon plus exacte, on peut dire qu’****une ACL est un moyen de sélectionner du trafic*** *pour en faire quelque chose, et ce quelque chose peut être du filtrage.*

**Différence entre liste d'accès standard et liste d'accès étendue :**

**Une liste d'accès standard** examinera seulement l'adresse IP source.   
  
**Une liste d'accès étendue** pourra examiner les adresses IP et les ports aussi bien source que destination, ainsi que le type de protocole (IP, ICMP, TCP, UDP). Par ailleurs, il sera possible de vérifier une partie des adresses avec un masque générique (wildcard mask).

Les paramètres contrôlés sont :

* Adresse source
* Adresse destination
* Protocole utilisé
* *Pour les protocoles utilisant les ports (TCP ou UDP)*
  + *Port source*
  + *Port destination*
* *D’autres options liées à d’autres protocoles (icmp echo-request, tcp established…)*

Les ACLs peuvent être appliquées sur le trafic entrant ou sortant. Il y a deux actions: soit le trafic est interdit, soit le trafic est autorisé.

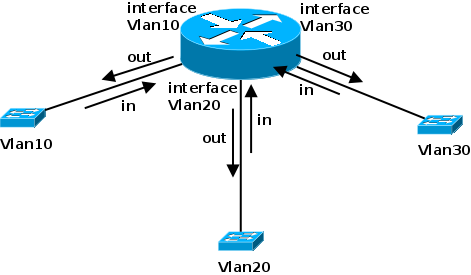
Les ACLs sont prises en compte de façon séquentielle. Il faut donc placer les instructions les plus précises en premier et l'instruction la plus générique en dernier.

Exemple :

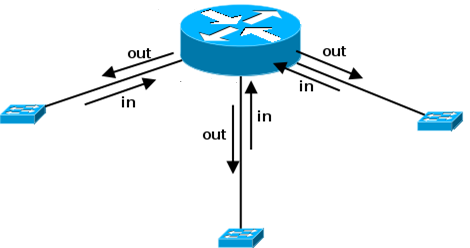
* Bloquer certains protocoles et autoriser le reste (quand peu de choses à bloquer)
* Autoriser certains protocoles et bloquer le reste (quand peu de choses à autoriser)

L’application d’une ACL se fait en entrée ou en sortie d’une interface (in/out).

**Dans le cas des « vlans » :** Pour le cas de figure des vlans il faut comprendre que le cœur de réseau est le centre de tous les « vlans », si on va du cœur de réseau vers un vlan c’est en « out » et quand on quitte un vlan pour aller vers un autre vlan on repasse par le cœur de réseau en « in ».



**Dans le cas de « ports » si on se trouve sur le cœur de réseau :**

****

Exemple de configuration pour un port physique (maquette) :

interface GigabitEthernet1/0/1

no switchport

ip address 172.20.254.2 255.255.255.248

ip access-group NOTELNET in

!

ip access-list extended NOTELNET

deny tcp any any eq telnet

permit tcp any any

!

Quelques opérateurs à connaitre pour faire des ACLs :

* **eq** : égal
* **neq** : différent
* **gt** : plus grand que
* **lt** : moins grand que

**Attention !** : Le mot « any » permet de cibler toutes les adresses IP(correspondance exacte : 0.0.0.0 255.255.255.255)

|  |  |
| --- | --- |
| Définir une ACL étendue | ip access-list extended <**100-199**> |
| Définir une ACL étendue nommée | Ip access-list extended <**NomAc**l> |
| Appliquer une ACL sur un port en entrée | conf t  int gi1/0/1  ip access-group <NomAcl> **in** |
| Appliquer une ACL sur un port en sortie | conf t  int gi1/0/1  ip access-group <NomAcl> **out** |
| Permet d’autoriser (permit) ou bloquer (deny) un type de trafic d’une source vers une destination | permit/deny ip/tcp/udp/icmp @source @MasqueSourceInverse @destination @MasqueDestinationInverse |
| Pour préciser le port source ou le port destination | Permit/deny <protocole> <source> <**OPTION**> <destination> <**OPTION**>  Le champ OPTION peut être remplacé par une sequence de type “eq x”, “gt y”, “lt z” ou “neq a”  Exemple : permit tcp any gt 1024 host 10.1.1.1 eq 80 |

**Quelques subtilités à connaitre :**

Dans le cas d**’ICMP** (ping) pour viser uniquement les « réponses » grâce à une ACL :

*permit / deny icmp @ip @masque @ip @masque* ***echo-reply***

Dans le cas de **TCP**, si on souhaite uniquement cibler le trafic des « réponses » :

permit / deny tcp @ip @masque @ip @masque ***established***

Dans le cas de **UDP**, si on souhaite filtrer, le plus simple est de suivre la convention qui dit que **les ports inférieurs à 1024 sont les services** (partie serveur théorique) et **les ports supérieurs à 1024 représentent les ports clients**.

permit / deny tcp @ip @masque ***gt 1024*** @ip @masque ***lt 1024***

Il est donc possible de bloquer les services ou les clients en suivant cette logique. (Voir opérateurs lt, gt, eq, neq).

Liste des ports d’applications courantes à connaitre:

|  |  |  |
| --- | --- | --- |
| **n°** | **type** | **description** |
| 20 | tcp | ftp-data - File Transfer Protocol [flux de données) |
| 21 | tcp | ftp - File Transfer Protocol (le flux de contrôle pour le transfert de fichiers), voir Diagramme des flux FTP (port 20 et 21) |
| 22 | tcp | SSH - Secure Shell |
| 23 | tcp | telnet |
| 25 | tcp | smtp - Simple Mail Transfer Protocol |
| 53 | udp/tcp | domain - Domain Name System (DNS) |
| 67 | udp | bootps - Bootstrap Protocol Server, ce port est aussi utilisé par une extension de bootp : DHCP, pour la recherche d'un serveur DHCP |
| 68 | udp | bootpc - Bootstrap Protocol Client, ce port est aussi utilisé par une extension de bootp : DHCP, pour le dialogue entre le serveur DHCP et le client (attribution d'un bail pour une adresse IP) |
| 69 | udp | tftp - Trivial File Transfer Protocol |
| 80 | tcp | www-http - World Wide Web HTTP |
| 110 | tcp | pop3 - Post Office Protocol - Version 3 |
| 115 | tcp | sftp - Simple File Transfer Protocol |
| 123 | udp | ntp - Network Time Protocol |
| 137 | tcp | netbios-ns - NETBIOS Name Service |
| 138 | tcp | netbios-dgm - NETBIOS Datagram Service |
| 139 | tcp | netbios-ssn - NETBIOS Session Service |
| 143 | tcp | imap2, imap4 - Internet Message Access Protocol v4 |
| 161 | udp | SNMP - Simple Network Management Protocol |
| 389 | tcp | Lightweight Directory Access Protocol (LDAP) |
| 443 | tcp | https |
| 445 | tcp/udp | microsoft-ds SMB, anciennement appelé CIFS (Common Internet File System) |
| 465 | tcp | smtp sécurisé (ssl) (non officiel) |
| 636 | tcp | LDAP encapsulé dans SSL/TLS |
| 993 | tcp | imap sécurisé (ssl) |
| 995 | tcp | pop3 sécurisé (ssl) |
| 1194 | udp/tcp | OpenVPN |
| 3389 | tcp | Microsoft Terminal Server (RDP) |
| 5900 | tcp | VNC Server |
| 8080 | tcp | http alternatif (webcache) *=> souvent utilisé comme port de proxy HTTP* |