

# CISCO

## LES ACL: ACCESS CONTROL LIST

### Objectifs

- Assimiler le concept général d'une ACL
- Comprendre le concept de Wildcard et savoir le calculer
- Interroger un appareil Cisco pour connaître les ACLs en place
- Savoir autoriser tout un réseau
- Savoir autoriser uniquement un hôte



QU'EST-CE QU'UNE  
ACL?

# LE CONCEPT GÉNÉRAL D'ACL

- **ACL Access Control List :**
  - **Système d'autorisations de paquets**, *ports, protocoles...*
  - **Plusieurs règles possibles dans une seule ACL**, *l'ACL est une liste !*
  - Chacune **vérifiée dans un ordre sé-quentiel !**
  - Au premier paquet matchant avec une règle de filtrage, qu'il soit accepté ou rejeté, les autres règles ne sont mêmes pas testées !

# LE CONCEPT GÉNÉRAL D'ACL

## 6 types d'ACL :

- ACL **Standard**
- ACL **Étendues**
- ACL **Dynamiques**
- ACL **Réflexives**
- ACL **basée sur le temps**
- ACL **basée sur le contexte (CBAC)**





POUR L'INSTANT NOUS NOUS  
INTÉRESSONS AUX  
ACL STANDARD

# LES ACL STANDARD

- Elles peuvent être **nommées** et/ou **numérotées**
- Le **numéro** va déterminer le **type** (*Standard ou Étendue*)
  - De **1 à 99**
  - Ou de **1300 à 1999**
- Une ACL par interface, ACL contenant plusieurs règles, dans chaque règle : un seul protocole

COMMENÇONS  
PAR LES ACLS  
«NUMÉROTÉES»

# LES ACL STANDARD

- La mise en place d'une ACL se fait en **3 étapes** :
  - **Création d'une ACL**
  - **Définir une ou plusieurs règles dans cette ACL**
  - **Appliquer l'ACL à une interface !**



# VOIR LES ACL ET EN CRÉER UNE :

**TOUJOURS Vérifier les ACL déjà en place :**

```
Router# show access-list
```

**Création d'une ACL avec une instruction pour un réseau IP:**

1

2

```
Router# access-list {numéro} {permit/deny} {préfixe} [wildcard masking] [log]
```

*préfixe* = adresse réseau

*log* = garde une trace de toutes les communications ayant « matché » la règle, facultatif

# ATTENTION!

**Dans la diapo précédente, nous avons...  
... réuni les étapes 1 et 2 en une seule commande !**

Séparer ces étapes est plus simple pour débiter avec les ACLs !

# COMME CECI :

①

```
Routeur(config)#ip access-list standard 1
```

②

```
Routeur(config-std-nacl)#permit 192.168.0.0 0.0.0.255
```



POURQUOI 0.0.0.255  
ET NON PAS  
255.255.255.0?

# LE WILCARD MASKING

- **Le Wildcard :**
  - Exact inverse du masque !

Exemples :

MASQUE	WILDCARD
255 . 0 . 0 . 0	0 . 255 . 255 . 255
255 . 255 . 255 . 0	0 . 0 . 0 . 255
255 . 255 . 255 . 248	0 . 0 . 0 . 7

# APPLIQUER L'ACL À UNE LIGNE OU À INTERFACE

3

Si je veux **appliquer mon ACL pour l'accès à une ligne**

```
Routeur(config)# line console 0 / line vty 0 - 15  
Routeur(config-line)# access-class numero/nom in/out
```

Si je veux **appliquer mon ACL pour l'accès à une interface**

```
Routeur(config)#interface gigabitEthernet 0/0  
Routeur(config)#ip access-group numero/nom in/out
```



UN EXEMPLE  
CONCRET?

# CRÉER UNE RÈGLE D'AUTORISATION :

## Création d'une ACL qui autorise tous les hôtes d'un réseau

1 2 `# access-list 1 permit 172.16.0.0 0.0.255.255`

## Attribution de cet ACL à l'interface Gigabit, en entrée

3 `Routeur(config)#interface gigabitEthernet 0/0  
Routeur(config-if)#ip access-group 1 in`





MAINTENANT LES  
ACLS «NOMMÉES»

# VOIR LES ACL ET CRÉER UNE AUTORISATION :

**TOUJOURS Vérifier les ACL déjà en places :**

```
Router# show access-list
```

**Création d'une ACL avec une instruction pour un réseau IP:**

1

2

```
Router# ip access-list standard NOM {permit/deny} {préfixe} [wildcard masking]
```

# ATTENTION!

**Vous noterez que dans le cas d'une ACL nommée...**

**On a du préciser « ip » ainsi que le mot « standard »**

Une gestion par nom est plus simple pour débiter avec les ACLs !

# ATTENTION!

A jongler entre numéros et noms.. on peut se retrouver avec des ACLs identiques !

```
R1#show access-lists
Standard IP access list 1
 10 permit 192.168.0.0, wildcard bits 0.0.0.255
 20 permit 192.168.1.0, wildcard bits 0.0.0.255
 30 deny 192.168.0.0, wildcard bits 0.0.3.255
 40 permit any
Standard IP access list monACL
 10 permit 192.168.0.0, wildcard bits 0.0.0.255
 20 permit 192.168.1.0, wildcard bits 0.0.0.255
 30 deny 192.168.0.0, wildcard bits 0.0.3.255
 40 permit any
```

ET  
AUTORISER/REJETER  
UN HÔTE  
PRÉCIS ?

# ACL SUR UN HÔTE PRÉCIS

Définir une ACL qui autorise un hôte en particulier :

```
Routeur(config)#access-list 1 permit host adresse@IP
```

Définir une ACL qui refuse un hôte en particulier :

```
Routeur(config)#access-list 1 deny host adresse@IP
```

# NOTE:

On peut désigner une machine précise en mettant le wildcard d'un masque de 32 bits, en enlevant le mot « host »

```
Routeur(config)#access-list 1 permit 172.16.0.54 0.0.0.0
```

*Mais c'est quand même plus simple en utilisant la méthode avec  
« host »...*



ET  
AU CONTRAIRE,  
AUTORISER ET/OU  
INTERDIRE «TOUT»?



# ACL DÉSIGNANT «PEU IMPORTE»

Nous devons utiliser le terme « **any** »

```
Routeur(config)#access-list 1 permit any
```

Le terme « any » remplace « 0.0.0.0 255.255.255.255 »

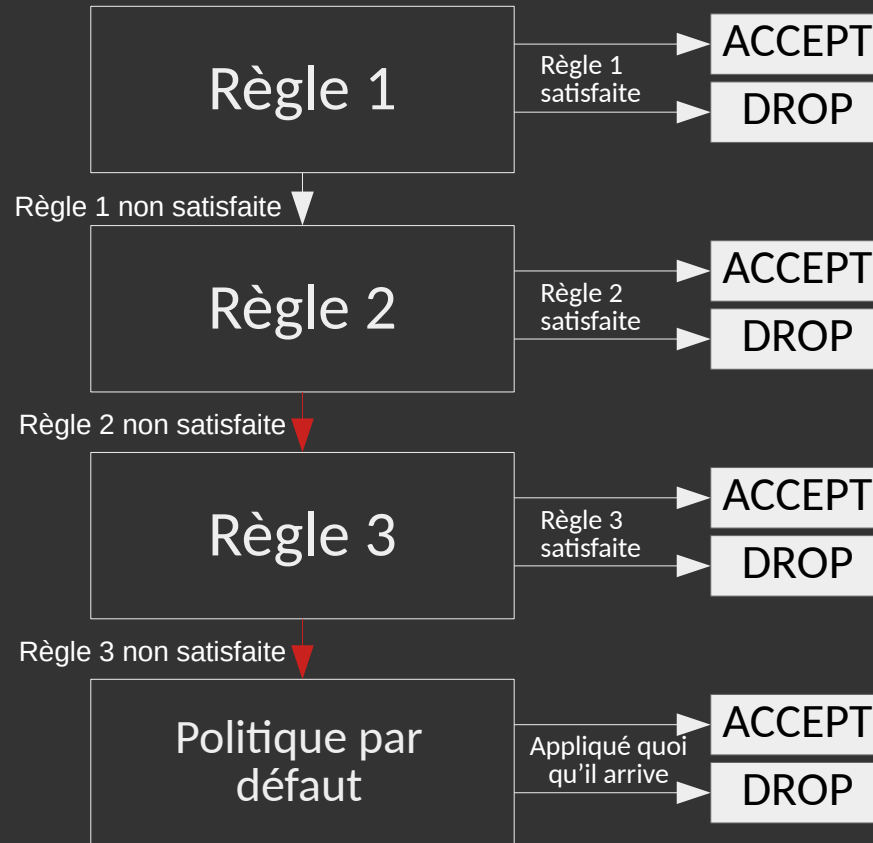


ATTENTION !

# ATTENTION AVEC LES ACL

- L'**ordre de** la création des **règles et/ou ACL** est **DÉTERMINANT**
- Une nouvelle règle est forcément ajoutée à la fin de la liste
- Il faut donc BIEN PRÉVOIR À L'AVANCE sous forme de schéma...
- ...les ACL à définir
- Parce qu'il est très difficile de supprimer une instruction particulière !

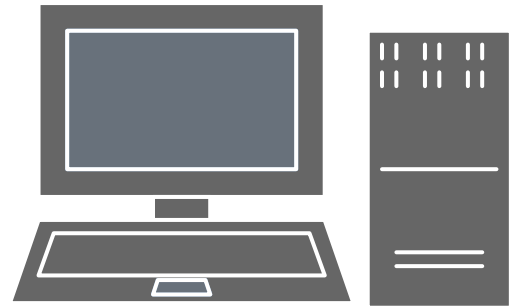
# TRAITEMENT DES RÈGLES DE FILTRAGE



# ATTENTION AVEC LES ACL

## Pour toute modification ...

- Il est préférable d'utiliser des éditeurs de texte
- On y recopie les ACL qu'on souhaite garder...
- ...On modifie celles le nécessitant...
- ...Sur le routeur, on supprime les ACL...
- ...**Puis on implante notre fichier texte avec nos nouvelles ACL**



RAPPEL !!  
LES ACLS SONT DES  
FILTRES SÉQUENTIELS !

# BIAIS DE MISE EN PLACE D'ACL :

## Que se passe-t-il ici ?

*access-list 1 permit any*

*access-list 1 deny host 10.1.1.1*

*access-list 1 deny any*



ET AUTORISER  
UN PROTOCOLE EN  
PARTICULIER ?



C'EST LE MOMENT  
DE PARLER DES  
ACL  
ÉTENDUES!

# ACL ÉTENDUES

- **ACL standard :**
  - **Filtre depuis une IPv4 source uniquement...**  
Donc très limité !
- **ACL étendue :**
  - **Filtre depuis une IP source vers une IP de destination !**
    - Ou d'un réseau source vers un réseau de destination...
  - **Peut filtrer à différents niveaux de protocoles !**

# ACL ÉTENDUES

- Elles peuvent être également **nommées** et/ou **numérotées**
- Le numéro va déterminer **le type**
  - De **100 à 199**
  - Ou de **2000 à 2699**
- Une ACL par interface, ACL contenant plusieurs règles, dans chaque règle : un seul protocole

# AUTORISER/RESTREINDRE UN PROTOCOLE

```
# access-list 100 permit tcp 172.16.0.0 0.0.255.255
```

- On peut soit mettre le nom du protocole...
- ...soit le numéro du protocole (*entre 1 et 255*)
  - Ces numéros sont propres à Cisco et n'ont **AUCUN RAPPORT** avec un numéro de port !

# QUELQUES N° DE PROTOCOLE POUR LES ACLS

Protocole	N° utilisable dans une règle pour désigner ce protocole
<b>ICMP</b>	<b>1</b>
<b>TCP</b>	<b>6</b>
<b>UDP</b>	<b>17</b>
IPv6	41
GRE	41
<b>EIGRP</b>	<b>88</b>
<b>OSPF</b>	<b>89</b>
PIM	103



ET SI LE PROTOCOLE N'EST  
PAS DANS LA LISTE DE  
CISCO... OU MÊME ASSIGNÉ À  
UN PORT STANDARDISÉ ?

# LES OPÉRANDES

**On va utiliser des « opérandes »**

*gt : greater than - supérieur à*

*eq : equal to - égal à*

*lt : less than - inférieur à*

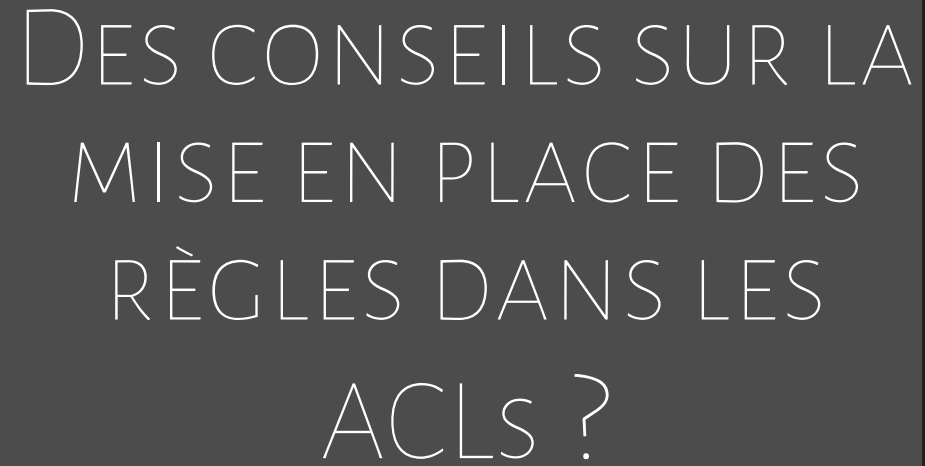
*nq : not equal - est différent de*

# EXEMPLE D'UTILISATION D'OPÉRANDE

```
# access-list 100 deny tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255  
eq 22
```

- Ici on définit un protocole de transport
- un réseau **source**
- un réseau de **destination**
- Un port précis correspondant au SSH (*tcp + port 22 = SSH*)





DES CONSEILS SUR LA  
MISE EN PLACE DES  
RÈGLES DANS LES  
ACLs ?

# CONSEILS D'IMPLEMENTATION ACL

**Les ACLs étendues** doivent être placées **au plus près de la source**

*(au niveau de l'interface)*

**Les ACLs standards** doivent être placées **au plus près de la destination**

*(puisqu'elles ne vérifient que la source)*



# CONSEILS D'IMPLÉMENTATION DE RÈGLES

**Placez les règles les plus précises en début d'ACL et les plus générales à la fin !**



ET  
POUR SUPPRIMER UNE  
RÈGLE..  
OU TOUTE UNE ACL ?

# ANNULER NOS ACTIONS

Supprimer une ACL toute entière :

```
Routeur(config)#no access-list 1 ...
```

Supprimer une règle d'une ACL (**possible UNIQUEMENT dans les ACL nommées**) :

```
Routeur(config)#access-list standard|extended NOM
```

```
Routeur(config)#no permit|deny ...
```

LES AUTRES TYPES  
D'ACLS (*DYNAMIQUES, CBAC*) NE  
SONT PAS L'OBJET DE CE  
COURS.