Cours 35 : Extended Access Control Lists

Dans ce cours nous allons continuer le cours sur les ACLs mais cette fois sur les Extented ACLs. Ce que nous avons vu à propos des ACLs dans le cours précédent est équivalent pour les Extended ACLs, la seule différence est que les les Extended ACLs peuvent ajouter plus de filtres par rapport aux ACLs Standard qui peut se baser uniquement sur l'adresse IP source d'un Paquet.

Durant ce cours nous allons donc voir comment configurer d'une autre manière les Numbered ACLs (ce qui peut s'appliquer pour les ACL standard et Extended), nous verrons également comment modifier une ACLs déjà configuré et en dernier temps nous verrons comment fonctionne les Extended Numbered et Named ACLs

Dans le cours précédent nous avons vu que les Numbered ACLs sont configurés en mode « global config »

```
R1(config)# access-list 1 deny 192.168.1.1
R1(config)# access-list 1 permit any
```

Nous avons également appris que les « Named ACLs » (ACLs només) sont configurés avec des sous commandes dans un mode de config différent.

```
R1(config)# ip access-list standard BLOCK_PC1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

Seulement aussi dans des IOS moderne il est aussi possible de configurer des numbered ACLs de la même manière que les Named ACLs comme dans cette cette exemple ci dessous configuré de la même manière que l'exemple ci dessus :

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

Ce sont juste deux manière différentes de configurer des Numbered ACLs, si l'on lance la commande « running-config » l'ACL apparaîtra comme si elle avait été configuré en utilisant la méthode traditionnel comme on peut le voir sur les commandes suivantes

```
R1(config)#ip access-list standard ?
  <1-99>        Standard IP access-list number
  <1300-1999>   Standard IP access-list number (expanded range)
  WORD          Access-list name

R1(config)#ip access-list standard 1
R1(config-std-nacl)#deny 192.168.1.1
R1(config-std-nacl)#permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 permit any
R1(config-std-nacl)#
```

Les avantages du mode de configuration Named ACL sont les suivant :
1. Il est possible de facilement supprimer une entrée dans l'ACL avec la valeur : « no » avant la commande. Voici comment cela fonctionne :

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#no 30
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
```

Voici en comparaison les commandes à lancer lorsque l'on utilise la méthode Standard Numbered ACLs :

```
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.3.0 0.0.0.255
access-list 1 permit any
R1(config)#no access-list 1 deny 192.168.3.0 0.0.0.255
R1(config)#do show access-lists
R1(config)#do show running-config | section access-list
R1(config)#
```

On peut voir ici que l'on a utilisé la même même méthode qu'auparavant pour supprimer l'entrée, mais que cette fois lorsque l'on veut afficher les ACLs avec la commande « show acces-lists » il n'y a aucun résultat. La raison est que ce n'est pas que l'entrée qui a été supprimé mais l'ACL entière. Lorsque l'on configure ou modifie des numbered ACLs depuis le mode global config mode, on ne peut pas seulement supprimer une entrée de manière individuel, mais on peut supprimer l'ACL en entier.

Ceci était le premier avantage d'utiliser le mode Named ACL.

2. Il y a un autre avantage qui est que l'on peut insérer de nouvelles entrées entre d'autres entrées en spécifiant le numéro de séquence comme on peut le voir dans l'exemple ci dessous :

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#30 deny 192.168.2.0 0.0.0.255
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.2.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.2.0 0.0.0.255
access-list 1 permit any
```

dans l'exemple l'entrée numéro 30 a été ajouté.
Il y a également la possibilité de reséquencer pour modifier l'ACL. La commande est :
« **ip access-list resequence** acl-id starting-seq-num increment »
Voici un exemple pour mieux comprendre :

```
R1(config)#do show access-lists
Standard IP access list 1
    1 deny   192.168.1.1
    3 deny   192.168.3.1
    2 deny   192.168.2.1
    4 deny   192.168.4.1
    5 permit any
R1(config)#
R1(config)#ip access-list resequence 1 10 10
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.3.1
    30 deny   192.168.2.1
    40 deny   192.168.4.1
    50 permit any
```

On peut voir ici que la commande
qui a été utilisé a permis de réorganiser l'ACL en commençant par la première entrée qui est : 1 qui a pris pour valeur 10 et sur laquelle est implémenté à chaque entrée la
valeur de 10 en plus.
On voit le résultat avec la commande « show access-lists »

Maintenant que nous avons vu les avantages d'utiliser les Named ACLs nous allons voir comment fonctionne un Extended ACLs
Comme précisé lors de l'introduction, le fonctionnement des Extended ACLs est le même que les Standard ACLs elles peuvent être numbered ou named comme les ACLs standard.
- Les ACLs Numbered utilisent les classements suivants : 100 - 199, 2000 - 2699
- Les Extended ACLs s'activent de la même manière que les Standard ACL du haut vers le bas.
La différence avec les Standard ACLs est qu 'elles peuvent utiliser plus de paramètres de filtrage, donc elles sont plus complexe et plus précises que les ACLs Standard.

Nous nous baserons sur les paramètres principaux qui sont : la couche 4 protocol/port, l'adresse IP source, et l'adresse de destination.

Pour configurer une Extended Numbered ACL depuis le mode Global Config la commande est :

```
R1(config)#access-list number [permit | deny] protocol src-ip dest-ip
```

il faut s'assurer que le nombre soir bien compris dans le classement précisé auparavant
(100 - 199, 2000 – 2699)

ensuite on peut lancer la commande :

```
R1(config-ext-nacl)#[seq-num] [permit | deny] protocol src-ip dest-ip
```

Voici un exemple de la configuration d'une Extended ACL

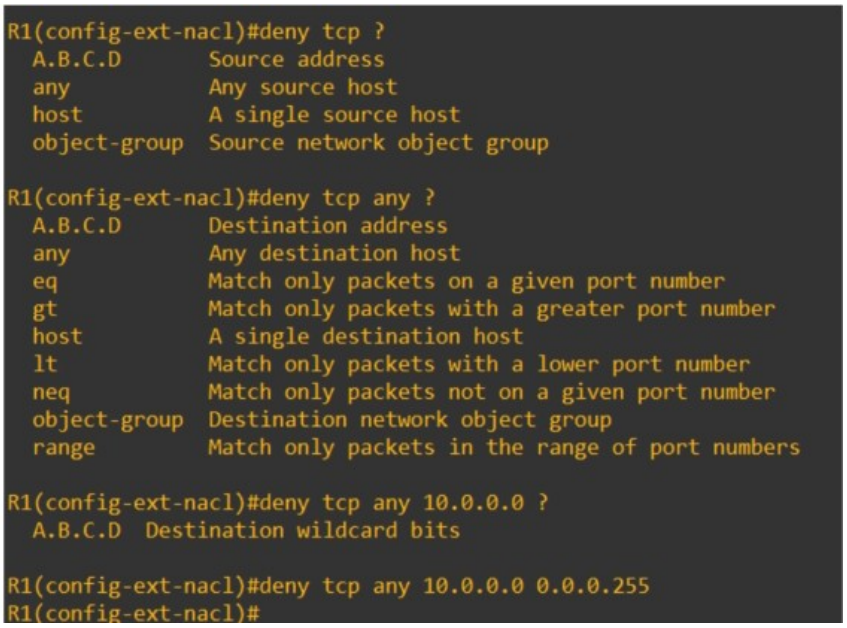
```
R1(config)#ip access-list extended EXAMPLE
R1(config-ext-nacl)#deny ?
  <0-255>        An IP protocol number
  ahp            Authentication Header Protocol
  eigrp          Cisco's EIGRP routing protocol
  esp            Encapsulation Security Payload
  gre            Cisco's GRE tunneling
  icmp           Internet Control Message Protocol
  igmp           Internet Gateway Message Protocol
  ip             Any Internet Protocol
  ipinip         IP in IP tunneling
  nos            KA9Q NOS compatible IP over IP tunneling
  object-group   Service object group
  ospf           OSPF routing protocol
  pcp            Payload Compression Protocol
  pim            Protocol Independent Multicast
  sctp           Stream Control Transmission Protocol
  tcp            Transmission Control Protocol
  udp            User Datagram Protocol
```

On peut voir que lorsque l'on veut lancer la commande deny avec un « ? » pour afficher les options on peut voir une liste des protocol.
En toute première ligne on peut voir le Protocole IP, cela permet d'identifier le numéro protocole qui est encapsulé dans l'entête IP comme TCP ou UDP. On peut aussi utiliser directement le nom du protocole parmi ceux listés.

Si l’on veut utiliser le numéro du protocole voici quelques exemples :
1 : ICMP
6 : TCP
17 : UDP
88 : EIGRP
89 : OSPF

Donc il possible par exemple de bloquer des messages OSPF sur une interface par exemple ou bien renier un paquet ICMP pour bloquer le ping.

Voyons un exemple pour voir comment ajouter une adresse IP source et une adresse IP de destination à cette entrée d’ACL.



Dans l’exemple ci dessus a été sélectionné le protocole TCP. Donc n’importe quelle paquet IP avec un segment TCP va correspondre à cette partie de l’entrée.
Il reste tout de même nécessaire de spécifier l’adresse IP source et de destination pour les faire correspondre.
Il est aussi à noter que dans l’Extended ACLs pour spécifier la source ou destination /32 il faut utiliser l’option **host** ou spécifier le masque de sous réseau. On ne peut pas juste écrire l’adresse sans ces autres paramètres, avec les ACLs Standard c’est possible mais pas avec les Extended.
C’est pour cela que dans l’exemple a été utilisé l’option « any » pour faire correspondre toutes les adresses IP source.
A la suite a été spécifié l’adresse IP de destination, il y a plusieurs options possible mais nous allons voir surtout les options « Destination Adresse » « Any » « Host ».
Dans l’exemple a été spécifié l’option avec l’adresse de destination 10.0.0.0 et le masque de sous réseau /24.

Nous allons à présent voir plusieurs exemples de configuration d'ACLs :
1. Permettre tout le trafique :
```
R1(config-ext-nacl)#permit ip any any
```

2. Empêcher 10.0.0.0/16 d'envoyer le trafique UDP à 192.168.1.1/32
```
R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1
```

3. Empêcher 172.16.1.1/32 de pinguer l'host dans 192.168.0.0/24
```
R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.0.0 0.0.0.255
```

Lorsque l'on veut faire correspondre TCP/UDP, on peut optionnellement spécifier la source et/ou le port de destination à faire correspondre.
La commande est la suivante :
« `R1(config-ext-nacl)#`**`deny tcp`** `src-ip dest-ip` »

Si l'on veut spécifier le port source ou de destination il faut le spécifier après l'adresse IP source.

« `R1(config-ext-nacl)#`**`deny tcp`** `src-ip` **`eq`** `src-port-num eq dest-ip` »

Les différentes options possible qui peuvent remplacer « eq » sont les suivantes :
- « `eq 80` » (equal) c'est égal au port 80 donc cela va faire correspondre la source TCP du port 80.
- « `gt 80` » (greater than). Par exemple gt 80 va faire correspondre tous les ports qui ont une valeur supérieur à 80.
- « `lt 80` » (less than) pour faire correspondre les ports inférieurs à 80
- « `neq 80` » (not equal) pour faire correspondre tous les autres port à l'exception du port 80.
- « `range 80 100` » pour faire correspondre le classement des ports compris entre 80 et 100.

Voici une liste du nom des différents protocoles utilisés :

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Il est à noter que lorsque l'on spécifie le protocole, l'adresse IP source, l'adresse Ip de destination, etc. Le paquet doit correspondre à toutes les valeurs pour correspondre l'entrée de l'ACL. Même si cela correspond à tous les autres paramètres à l'exception d'un seul le paquet ne correspondra pas à cette entrée de l'ACL.

Donc en résumé l'Extended ACL permet d'être très précis dans le filtrage des paramètres.

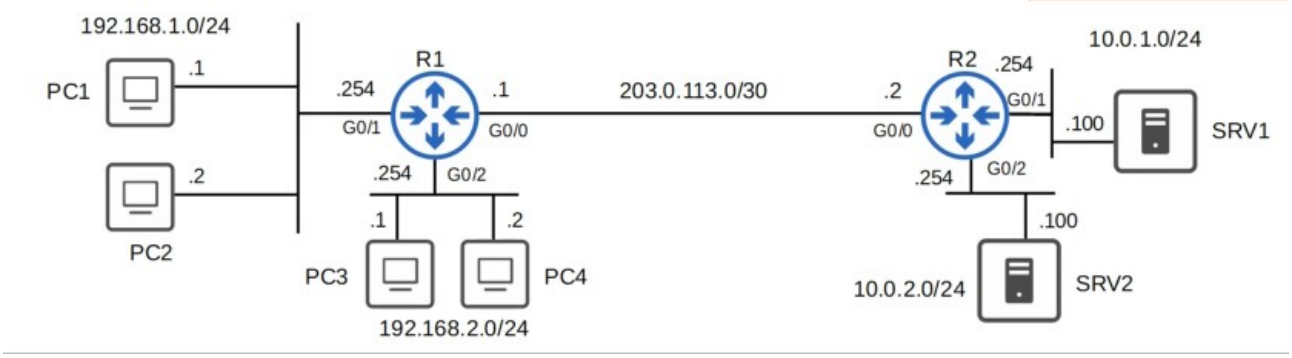
Voici quelques exemples d'ACLs :
1. permettre le trafique de 10.0.0.0/16 d'accéder au serveur à 2.2.2.2/32 en utilisant HTTPS.
```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2
0.0.0.0 eq 443
```

2. Empêcher tous les hôtes d'utiliser le port source UDP de 20000 à 30000 d'accéder au serveur 3.3.3.3/32
```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Permettre à l’hôte dans 172.16.1.0/24 d’utiliser le port source TCP supérieur à 9999 d’accéder à tous les ports TCP du serveur 4.4.4.4/32 excepté le port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host
4.4.4.4 neq 23
```

A présent nous allons configurer le réseau suivant pour que les conditions édictés ci dessous sous bien respectés.



Les conditions de configurations sont les suivantes :
- Les hôtes dans 192.168.1.0/24 ne peuvent pas utiliser HTTPS pour accéder au SRV1
- Les hôtes dans 192.168.2.0/24 ne peuvent pas accéder à 10.0.2.0/24
- Les hôtes dans 192.168.1.0/24 ou 192.168.2.0/24 peut pinger 10.0.1.0/24 ou 10.0.2.0/24

Voici les commandes que nous allons utiliser pour que la première partie de conditions :

```
R1(config)#ip access-list ectended HTTP_SRV1
R1(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100
eq 443
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/1
R1(config-if)#ip access-group HTTP_SRV1 in
```

Il est à noter que les Extended ACLs doivent s’appliquer le plus proche possible de la source, pour limiter le trajet du trafique avant qu’il ne soit renié.
(Standard ACLs sont plus spécifiques, donc s’ils s’appliquent plus proche de la source il y a un risque de bloquer plus de trafique que voulu).

Pour la deuxième partie des conditions voici les commandes à utiliser :

```
R1(config)#ip access-lis extended BLOCK_10.0.2.0/24
R1(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 10.0.2.0
0.0.0.255
R1(config-ext-nacl)#permit ip any any
```

Pour la troixième partie des conditions voici les commandes à utiliser :

```
R1(config)#ip access-list extended BLOCK_ICMP
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.1.0
0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.2.0
0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.2.0 0.0.0.255 10.0.1.0
0.0.0.255
R1(config-ext-nacl)#permit ip nay any
R1(config-ext-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_ICMP out
```

Voici les résultats des commandes pour afficher les commandes :

```
R1#show access-lists
Extended IP access list BLOCK_10.0.2.0/24
    10 deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
    20 permit ip any any
Extended IP access list BLOCK_ICMP
    10 deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
    20 deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
    30 deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
    40 permit ip any any
Extended IP access list HTTP_SRV1
    10 deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
    20 permit ip any any
```

```
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 203.0.113.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BLOCK_ICMP
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
```