

## TD 2 : BGP AS Path Filter

### Introduction

Dans ce TD, vous allez apprendre comment appliquer un filtrage de chemin BGP au sein d'un AS. En effet, grâce à cette technique nous pouvons autoriser ou refuser les préfixes de certains systèmes autonomes. Vous pouvez utiliser une telle technique pour par exemple :

- N'acceptez que les préfixes des SA directement connectés
- N'acceptez que les préfixes des SA directement connectés et de ceux se trouvant juste à un saut après.
- Refuser certains systèmes autonomes de transit
- Etc.

Pour créer des règles comme les exemples ci-dessus, nous avons besoin d'une manière flexible afin que nous puissions "faire correspondre" des règles en fonction des systèmes autonomes. Cela peut être fait avec des expressions régulières. En effet, ces expressions sont souvent utilisées pour la manipulation ou le filtrage des routes BGP. Avant de voir quelques exemples d'expressions régulières utiles, voyons d'abord les différents caractères que nous pouvons utiliser :

|    |   |
|----|---|
| ?  | répète le caractère précédent une ou zéro fois.                                   |
| *  | répète le caractère précédent zéro ou plusieurs fois.                             |
| +  | répète le caractère précédent une ou plusieurs fois.                              |
| ^  | correspond au début d'une chaîne.   |
| \$ | correspond à la fin d'une chaîne.   |
| [] | est une plage.  |
| -  | correspond à l'espace entre les numéros AS ou à la fin de la liste AS PATH.       |
| \\ | est un caractère d'échappement. Vous en aurez besoin pour les confédérations BGP. |

Pour voir des exemples concrets, vous allez faire des tests sur un serveur miroir BGP. Un serveur miroir est un routeur sur Internet qui dispose d'une table de routage Internet (complète). Vous pouvez utiliser telnet vers un de ces routeurs et utiliser les commandes show pour afficher la table BGP. C'est un excellent moyen de pratiquer les expressions régulières car il y a beaucoup de préfixes avec lesquels on peut tester.

Vous pouvez trouver un serveur miroir sur <https://www.bgp4.as/looking-glasses>. Pour les exemples ci-après, on va choisir le serveur **route-server.opentransit.net** qui fait partie de l'AS 5511 :

```
OAKRS1#show ip bgp
BGP table version is 3695525243, local router ID is 204.59.3.38
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|     | Network    | Next Hop        | Metric | LocPrf | Weight | Path                   |
|-----|------------|-----------------|--------|--------|--------|------------------------|
| r>i | 0.0.0.0    | 193.251.245.7   |        | 100    | 0      | i                      |
| * i | 1.0.0.0/24 | 193.251.142.12  |        | 150    | 0      | 13335 i                |
| * i |            | 193.251.150.138 |        | 150    | 0      | 13335 i                |
| *>i |            | 193.251.150.108 |        | 150    | 0      | 13335 i                |
| * i |            | 193.251.245.37  |        | 150    | 0      | 13335 i                |
| * i |            | 193.251.150.8   |        | 150    | 0      | 13335 i                |
| * i | 1.0.4.0/22 | 195.219.87.114  | 100    | 85     | 0      | 6453 7545 2764 38803 i |
| * i |            | 193.251.143.44  | 100    | 85     | 0      | 174 7545 2764 38803 i  |
| * i |            | 154.54.10.105   | 100    | 85     | 0      | 174 7545 2764 38803 i  |
| * i |            | 195.12.254.102  | 100    | 85     | 0      | 1299 7545 2764 38803 i |

Pour l'ensemble des exemples suivants, vous êtes invité à faire le test vous-même pour une meilleure compréhension.

**Exemple 1** : l'objectif ici est de n'afficher que les préfixes provenant directement de l'AS 3257. Ainsi tous les autres préfixes ne seront pas affichés. La commande sera :  
**show ip bgp regexp ^3257\$**

```
OAKRS1#show ip bgp regexp ^3257$
BGP table version is 3696334046, local router ID is 204.59.3.38
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|     | Network        | Next Hop        | Metric | LocPrf | Weight | Path   |
|-----|----------------|-----------------|--------|--------|--------|--------|
| * i | 2.56.236.0/22  | 193.251.142.16  | 100    | 85     | 0      | 3257 i |
| * i |                | 193.251.245.37  | 100    | 85     | 0      | 3257 i |
| * i |                | 193.251.142.76  | 100    | 85     | 0      | 3257 i |
| *>i |                | 212.115.144.89  | 100    | 85     | 0      | 3257 i |
| * i |                | 212.115.169.185 | 100    | 85     | 0      | 3257 i |
| * i | 2.58.75.0/24   | 193.251.142.16  | 100    | 85     | 0      | 3257 i |
| * i |                | 193.251.245.37  | 100    | 85     | 0      | 3257 i |
| * i |                | 193.251.142.76  | 100    | 85     | 0      | 3257 i |
| *>i |                | 212.115.144.89  | 100    | 85     | 0      | 3257 i |
| * i |                | 212.115.169.185 | 100    | 85     | 0      | 3257 i |
| * i | 5.133.190.0/24 | 193.251.142.16  | 100    | 85     | 0      | 3257 i |
| * i |                | 193.251.245.37  | 100    | 85     | 0      | 3257 i |

L'expression régulière utilisée ici comporte le symbole ^ qui signifie qu'il s'agit du début de la chaîne et le \$ correspond à la fin de la chaîne. Avec la valeur 3257 entre les deux, cela veut dire qu'on veut seulement l'AS "3257".

**Exemple 2 :** afficher seulement les préfixes qui sont passés par l'AS 3257, voici comment : `show ip bgp regexp _60051_`

```
OAKRS1#show ip bgp regexp _60051_
BGP table version is 3696364763, local router ID is 204.59.3.38
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*>i  5.10.229.0/24  4.68.127.233      100      85      0 3356 60051 208293 203214 3194 i
* i    4.68.73.109      100      85      0 3356 60051 208293 203214 3194 i
* i    193.251.150.174  100      85      0 3356 60051 208293 203214 3194 i
* i    4.68.73.97      100      85      0 3356 60051 208293 203214 3194 i
* i    4.68.73.153     100      85      0 3356 60051 208293 203214 3194 i
* i    4.68.73.61      100      85      0 3356 60051 208293 203214 3194 i
* i  31.223.186.0/24  4.68.73.97      100      85      0 3356 60051 208293 203214 208365 i
* i    4.68.73.153     100      85      0 3356 60051 208293 203214 208365 i
* i    193.251.150.174  100      85      0 3356 60051 208293 203214 208365 i
*>i    4.68.127.233     100      85      0 3356 60051 208293 203214 208365 i
* i    4.68.73.61      100      85      0 3356 60051 208293 203214 208365 i
* i    4.68.73.109     100      85      0 3356 60051 208293 203214 208365 i
* i  37.98.227.0/24  4.68.73.97      100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.153     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.109     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.61      100      85      0 3356 60051 208293 203214 50710 50597 i
*>i    4.68.127.233     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    193.251.150.174  100      85      0 3356 60051 208293 203214 50710 50597 i
* i  37.98.228.0/24  4.68.73.97      100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.153     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.109     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    4.68.73.61      100      85      0 3356 60051 208293 203214 50710 50597 i
*>i    4.68.127.233     100      85      0 3356 60051 208293 203214 50710 50597 i
* i    193.251.150.174  100      85      0 3356 60051 208293 203214 50710 50597 i
```

L'expression régulière commence et se termine par un `_` cela correspond à l'espace entre les numéros de chemin AS. On n'utilise pas `^` ou `$` pour indiquer le début et la fin de la chaîne afin qu'il puisse y avoir autant d'AS que nous le souhaitons, tant qu'il passe par AS 60051, il correspondra.

**Exemple 3 :** afficher les préfixes provenant de l'AS 56203 (qui n'est pas forcément connecté à l'AS où depuis lequel on applique la commande). Pour cela la commande à utiliser est : `show ip bgp regexp _203214$`

```
OAKRS1#show ip bgp regexp _203214$
BGP table version is 3696380159, local router ID is 204.59.3.38
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|     | Network        | Next Hop        | Metric | LocPrf | Weight | Path                       |
|-----|----------------|-----------------|--------|--------|--------|----------------------------|
| * i | 37.236.0.0/16  | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| *>i |                | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| * i |                | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| * i | 37.236.7.0/24  | 193.251.150.76  | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| *>i |                | 193.251.248.0   | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 193.251.142.106 | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 195.12.254.102  | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 62.115.170.120  | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 195.12.254.154  | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 62.115.155.28   | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i | 37.236.8.0/24  | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| * i |                | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| *>i |                | 172.25.4.85     |        | 100    | 0      | 39386 208293 203214 ?      |
| * i | 37.236.10.0/24 | 193.251.142.106 | 100    | 85     | 0      | 1299 60051 208293 203214 i |
| * i |                | 193.251.150.76  | 100    | 85     | 0      | 1299 60051 208293 203214 i |

Le premier AS est toujours sur le côté droit, donc pour faire correspondre à ce qu'on cherche, nous terminons la chaîne par un \$ et mettons le numéro AS juste devant.

Le \_ correspondra à l'espace devant le numéro AS.

Un tel filtrage peut être utile si on souhaite bloquer les préfixes provenant d'un AS particulier et autoriser tout le reste.

**Exemple 4 :** Afficher les préfixes provenant de l'AS 3257 et de ses AS directement connectés, voici comment : `show ip bgp regexp ^3257_[0-9]*$`

```
OAKRS1#show ip bgp regexp ^3257_[0-9]*$
BGP table version is 3696430007, local router ID is 204.59.3.38
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|     | Network       | Next Hop        | Metric | LocPrf | Weight | Path          |
|-----|---------------|-----------------|--------|--------|--------|---------------|
| * i | 1.118.32.0/24 | 193.251.142.16  | 100    | 85     | 0      | 3257 137753 i |
| * i |               | 193.251.142.76  | 100    | 85     | 0      | 3257 137753 i |
| * i |               | 212.115.169.185 | 100    | 85     | 0      | 3257 137753 i |
| *>i |               | 212.115.144.89  | 100    | 85     | 0      | 3257 137753 i |
| * i |               | 193.251.245.37  | 100    | 85     | 0      | 3257 137753 i |
| * i | 2.17.116.0/22 | 193.251.142.76  | 100    | 85     | 0      | 3257 8708 i   |
| * i |               | 212.115.144.89  | 100    | 85     | 0      | 3257 8708 i   |
| * i | 2.20.96.0/19  | 193.251.142.76  | 100    | 85     | 0      | 3257 8708 i   |
| * i |               | 212.115.144.89  | 100    | 85     | 0      | 3257 8708 i   |
| * i | 2.22.72.0/22  | 193.251.245.37  | 100    | 85     | 0      | 3257 852 i    |
| * i |               | 193.251.142.76  | 100    | 85     | 0      | 3257 852 i    |

On commence par le ^3257 afin de n'accepter que les préfixes de l'AS 3257. Le \_ correspondra à l'espace et le [0-9] correspondra à n'importe quel caractère entre 0

et 9. Le \* signifie que nous répétons le dernier caractère (0-9). Cela signifie que AS1 correspondrait, mais aussi AS123 ou AS12345, etc. Le \$ à la fin garantira qu'un seul système autonome derrière AS 3257 est autorisé.

L'objectif des exemple précédents c'est de vous montrer comment exprimer avec une expression régulière un besoin précis de garder ou d'exclure des préfixes en fonction des numéros des AS traversés. Dans la suite vous allez voir comment appliquer ces expressions dans des filtres à appliquer sur des routeurs, permettant ainsi d'agir sur les informations échangées.

Ainsi, pour configurer un filtre AS sur un routeur Cisco IOS, vous pouvez le faire avec les commandes suivantes :

```
R(config)#ip as-path access-list 1 permit ^3257$  
  
R(config)#router bgp 1  
R(config-router)#neighbor X.X.X.X remote-as xyx  
R(config-router)#neighbor X.X.X.X filter-list 1 in
```

La 1ere ligne permet de déclarer la liste d'accès **as-path** qui fonctionne comme les listes d'accès normales. À savoir que la règle par défaut c'est un "deny any". Une fois la liste d'accès as-path crée (avec le numéro 1 ici), il faut l'appliquer au voisin BGP souhaité dans le sens entrant. Ainsi le routeur va filtrer les messages BGP reçu de ce voisin en appliquant l'expression régulière définie dans le filtre.

## Exercice 1 :

Donner la configuration à appliquer pour les cas suivants :

1. N'autoriser que les préfixes provenant de l'AS 3257
2. Autoriser uniquement les réseaux passant par l'AS 3257
3. Refuser les préfixes provenant de l'AS 56203 et autoriser tout le reste
4. Autoriser les préfixes de l'AS 3257 et de ses AS directement connectés mais refuser le reste

## Exercice 2 :

Vous travaillez pour une entreprise qui utilise BGP pour échanger des informations de routage entre ses sites. Vous souhaitez filtrer les routes entrantes de sorte que seuls les chemins AS qui passent par l'AS 65001 soient acceptés. Donner les étapes à suivre pour configurer ce filtrage AS Path.

## Exercice 3 :

- 1) Donner la configuration à appliquer pour n'autoriser que les préfixes provenant de l'AS 1234.
- 2) Comment configurer un filtre AS Path pour n'autoriser que les préfixes provenant des AS 65001 et 65002 ?
- 3) Donner la configuration à appliquer pour filtrer les préfixes BGP en fonction de leur AS Path et n'autoriser que les préfixes provenant des AS 65000 à 65500.
- 4) Comment configurer un filtre AS Path pour autoriser les préfixes provenant uniquement des AS 1234 et 5678, mais refuser tous les autres préfixes entrants ?
- 5) Donnez la configuration à appliquer pour n'autoriser que les préfixes provenant des AS 3257 et 65001 à 65003, tout en refusant tous les autres préfixes entrants.
- 6) Comment configurer un route-map pour autoriser les préfixes de l'AS 1234 et de ses AS directement connectés, mais refuser les préfixes provenant de l'AS 5678.