Contrôle des politiques BGP

SI-F

AfNOG 2014

Last updated 24 April 2013

Appliquer des politiques avec BGP

- Politiques-basées sur l'AS path, les communautés ou le préfixe
- Rejet/acceptation des routes sélectionnées
- Assigne des valeurs aux attributs pour influencer la sélection de chemin
- Outils:
  - Prefix-list (filtres les préfixes)
  - Filter-list (filtres les ASs)
  - Route-maps et communautés

Contrôle des politiques – Prefix List

- Filtre de préfixe par voisin
  - Configuration incrémentale
- En entrée ou en sortie (Inbound or Outbound)
- Basé sur l'identifiant des préfixes (notation adresse IPv4/masque, idem pour IPv6)
- L'utilisation des access-lists en Cisco IOS pour filtrer les préfixes est périmé depuis longtemps
  - **Fortement decouragé!**

Syntaxe pour les Prefix-list

- Syntaxe:

  `[no] ip prefix-list list-name [seq seq-value] permit|deny network/len [ge ge-value] [le le-value]`

  `network/len:` préfixe et longueur

  `ge ge-value:` “plus grand ou égal à”

  `le le-value:` “plus petit ou égal à”

- “ge” et “le” sont tous deux optionnels
  - Pour spécifier l’intervalle des longueurs de préfixes à faire correspondre pour les préfixes qui sont plus spécifiques que network/longueur
- Le numéro de séquence est optionnel

  `no ip prefix-list sequence-number` pour désactiver l’affichage des numéros de séquence

Prefix Lists – Exemples

- Rejeter la route par défaut

  `ip prefix-list EG deny 0.0.0.0/0`

- Permettre le préfixe 35.0.0.0/8

  `ip prefix-list EG permit 35.0.0.0/8`

- Rejeter le préfixe 172.16.0.0/12

  `ip prefix-list EG deny 172.16.0.0/12`

- Dans 192/8 permettre les prefixes jusqu'à /24

  `ip prefix-list EG permit 192.0.0.0/8 le 24`

  - Permet toutes les tailles de prefixes du bloc 192.0.0.0/8 **sauf** /25, /26, /27, /28, /29, /30, /31 et /32.

Prefix Lists – Exemples

- Dans 192/8 rejeter /25 et au dessus

  `ip prefix-list EG deny 192.0.0.0/8 ge 25`

  - Ceci rejete les tailles de prefixe /25, /26, /27, /28, /29, /30, /31 et /32 dans le bloc d'adresses 192.0.0.0/8.
  - Cela a le même effet que l'exemple précédent

- Dans 193/8 permettre les préfixes entre /12 et /20

  `ip prefix-list EG permit 193.0.0.0/8 ge 12 le 20`

  - Ceci rejette les préfixes de tailles /8, /9, /10, /11, /21, /22, … et plus dans le bloc d'adresses 193.0.0.0/8.

- Permet tous les préfixes

  `ip prefix-list EG permit 0.0.0.0/0 le 32`

  - 0.0.0.0 correspond à toutes adresses possibles, "0 le 32" correspond à toutes les longueurs de préfixes

Contrôle des politiques – Prefix List

- Exemple de configuration

```
router bgp 100
 network 105.7.0.0 mask 255.255.0.0
 neighbor 102.10.1.1 remote-as 110
 neighbor 102.10.1.1 prefix-list AS110-IN in
 neighbor 102.10.1.1 prefix-list AS110-OUT out
!
ip prefix-list AS110-IN deny 218.10.0.0/16
ip prefix-list AS110-IN permit 0.0.0.0/0 le 32
ip prefix-list AS110-OUT permit 105.7.0.0/16
ip prefix-list AS110-OUT deny 0.0.0.0/0 le 32
```

Contrôle des politiques – Filter List

- Filtre les routes sur base de l'AS path
  - Entrée ou sortie
- Exemple de configuration:

```
router bgp 100
 network 105.7.0.0 mask 255.255.0.0
 neighbor 102.10.1.1 filter-list 5 out
 neighbor 102.10.1.1 filter-list 6 in
!
ip as-path access-list 5 permit ^200$
ip as-path access-list 6 permit ^150$
```

Contrôle des politiques – Expressions Régulières

- Comme les expressions régulières Unix
  - . Correspond a un caractère
  - * N'importe quel nombre d'occurrences de l'expression précédente
  - + Au moins une occurrence de l'expression précédente
  - ^ Début de ligne
  - $ Fin de ligne
  - \ Echapper à une caractère d'expression régulière
  - _ Début, fin de ligne, espace, accolade
  - | Ou
  - () parenthèse pour contenir une expression
  - [] crochets pour contenir un intervalle (nombre d'occurrences)

Contrôle des politiques – Expressions Régulières

- Exemples simples

.*	tout correspond
.+	toute expression avec au moins un caractère
^$	routes locales à cet AS
_1800$	générée pas AS1800
^1800_	recue de AS1800
_1800_	via AS1800
_790_1800_	via AS1800 et AS790
_(1800_)+	plusieurs fois AS1800 en séquence (pour la AS-PATH prepend)
_\(65530\)_	via AS65530 (confédérations)

Contrôle des politiques – Expressions Régulières

- Exemples plus compliqués

^[0-9]+$	Correspond à AS_PATH de longueur 1
^[0-9]+_[0-9]+$	Correspond à AS_PATH de longueur 2
^[0-9]*_[0-9]+$	Correspond à AS_PATH de longueur 1 ou 2
^[0-9]*_[0-9]*$	Correspond à AS_PATH de longueur 0, 1 ou 2
^[0-9]+_[0-9]+_[0-9]+$	Correspond à AS_PATH de longueur 3
_(701|1800)_	Correspond à tout ce qui est passé par AS701 ou AS1800
_1849(_.+_)12163$	Correspond à AS12163 comme origine et qui passe par AS1849

Contrôle des politiques – Route Maps

- Une route-map est comme un “programme” pour IOS
- Avec des numéros de ligne, comme un programme
- Chaque ligne est un couple condition/action
- Le concept de base est:

  if *match* then do *expression* and exit

  else

  if *match* then do *expression* and exit

  else etc
- Le mot clé “continue” permet à l’ISPs d’appliquer plusieurs couples conditions/actions à la suite dans un route-map

Route Maps – Avertissements

- Lignes peuvent avoir plusieurs déclarations “set”
- Lignes peuvent avoir plusieurs déclarations “match”
- Ligne avec une seule déclaration match
  - Seuls les prefixes qui correspondent passent les autres sont jetés
- Ligne avec une seule déclaration set
  - Tous les préfixes correspondent et donc subissent le « set »
  - Les lignes suivantes de la route-map sont ignorées
- Ligne avec une déclaration match/set et pas de lignes suivantes
  - Seules les préfixes qui correspondent subissent le “set”, les autres préfixes sont jetés

Route Maps – Avertissements

- Exemple
  - Omittre la 3eme ligne ci-dessous signifie que les préfixes qui ne correspondent pas à list-one ou list-two sont jetés

```
route-map sample permit 10
 match ip address prefix-list list-one
 set local-preference 120
!
route-map sample permit 20
 match ip address prefix-list list-two
 set local-preference 80
!
route-map sample permit 30 ! Don't forget this
```

Route Maps – Matching prefixes

- Example Configuration

```
router bgp 100
 neighbor 1.1.1.1 route-map infilter in
!
route-map infilter permit 10
 match ip address prefix-list HIGH-PREF
 set local-preference 120
!
route-map infilter permit 20
 match ip address prefix-list LOW-PREF
 set local-preference 80
!
ip prefix-list HIGH-PREF permit 10.0.0.0/8
ip prefix-list LOW-PREF permit 20.0.0.0/8
```

Route Maps – AS-PATH filtering

- Example Configuration

```
router bgp 100
 neighbor 102.10.1.2 remote-as 200
 neighbor 102.10.1.2 route-map filter-on-as-path in
!
route-map filter-on-as-path permit 10
 match as-path 1
 set local-preference 80
!
route-map filter-on-as-path permit 20
 match as-path 2
 set local-preference 200
!
ip as-path access-list 1 permit _150$
ip as-path access-list 2 permit _210_
```

Route Maps – AS-PATH prepends

- Example configuration of AS-PATH prepend

```
router bgp 300
 network 105.7.0.0 mask 255.255.0.0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
 set as-path prepend 300 300
```

- Use your own AS number when prepending
  - Otherwise BGP loop detection may cause disconnects

Route Maps – Matching Communities

- Example Configuration

```
router bgp 100
 neighbor 102.10.1.2 remote-as 200
 neighbor 102.10.1.2 route-map filter-on-community in
!
route-map filter-on-community permit 10
 match community 1
 set local-preference 50
!
route-map filter-on-community permit 20
 match community 2 exact-match
 set local-preference 200
!
ip community-list 1 permit 150:3 200:5
ip community-list 2 permit 88:6
```

Community-List Processing

- Note:
  - When multiple values are configured in the same community list statement, a logical AND condition is created. All community values must match to satisfy an AND condition

```
ip community-list 1 permit 150:3 200:5
```

  - When multiple values are configured in separate community list statements, a logical OR condition is created. The first list that matches a condition is processed

```
ip community-list 1 permit 150:3
ip community-list 1 permit 200:5
```

Route Maps – Setting Communities

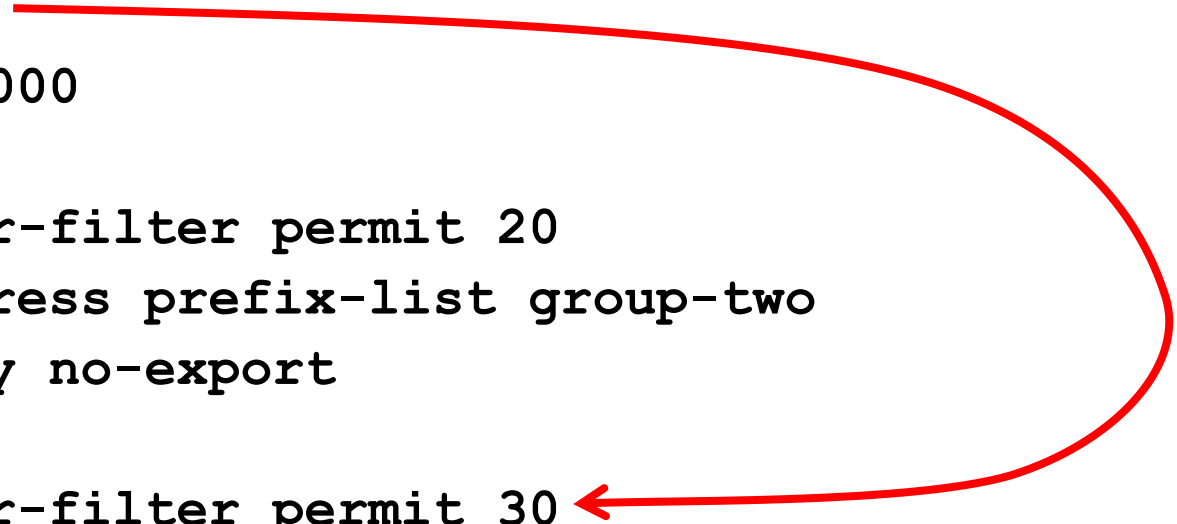
- Example Configuration

```
router bgp 100
 network 105.7.0.0 mask 255.255.0.0
 neighbor 102.10.1.1 remote-as 200
 neighbor 102.10.1.1 send-community
 neighbor 102.10.1.1 route-map set-community out
!
route-map set-community permit 10
 match ip address prefix-list NO-ANNOUNCE
 set community no-export
!
route-map set-community permit 20
 match ip address prefix-list AGGREGATE
!
ip prefix-list NO-ANNOUNCE permit 105.7.0.0/16 ge 17
ip prefix-list AGGREGATE permit 105.7.0.0/16
```

Route Map Continue

- Handling multiple conditions and actions in one route-map (for BGP neighbour relationships only)

```
route-map peer-filter permit 10
 match ip address prefix-list group-one
 continue 30
 set metric 2000
!
route-map peer-filter permit 20
 match ip address prefix-list group-two
 set community no-export
!
route-map peer-filter permit 30
 match ip address prefix-list group-three
 set as-path prepend 100 100
!
```



Order of processing BGP policy

- For policies applied to a specific BGP neighbour, the following sequence is applied:
  - For inbound updates, the order is:
    - Route-map
    - Filter-list
    - Prefix-list
  - For outbound updates, the order is:
    - Prefix-list
    - Filter-list
    - Route-map

Managing Policy Changes

- New policies only apply to the updates going through the router AFTER the policy has been introduced or changed
- To facilitate policy changes on the entire BGP table the router handles the BGP peerings need to be “refreshed”
  - This is done by clearing the BGP session either in or out, for example:

```
clear ip bgp <neighbour-addr> in|out
```

- Do NOT forget in or out — doing so results in a hard reset of the BGP session

Managing Policy Changes

- Ability to clear the BGP sessions of groups of neighbours configured according to several criteria
- **`clear ip bgp <addr> [in|out]`**

  `<addr>` may be any of the following

`x.x.x.x`	IP address of a peer
`*`	all peers
`ASN`	all peers in an AS
`external`	all external peers
`peer-group <name>`	all peers in a peer-group

BGP Policy Control

ISP Workshops