

Route Filter Operation



- **Routes to be filtered** can be specified with an Access Control List or Prefix List
- ACLs and Prefix Lists can optionally be nested inside a Route Map
- The filter is **applied** with a Distribute List
- (Prefix Lists can be applied directly to neighbors in BGP)

```
R1(config-router)#distribute-list ?
<1-199>          IP access list number
<1300-2699>     IP expanded access list number
WORD             Access-list name
gateway          Filter based on gateway
prefix           Filter prefixes in routing updates
route-map        Filter prefixes based on route-map
```

<https://t.me/learningnets>

Route Maps



- Route Maps provide 'if – then' programming logic
- They share many similarities with Access Control Lists:
 - They both match specified packets, and take an action of Permit or Deny
 - They are both read top down, and stop processing on the first match
 - They both have an implicit Deny at their end

```
route-map FILTER deny 10
  match ip address ACL-A
route-map FILTER permit 20
  match ip address prefix-list LIST-B
```

```
router ospf 1
  distribute-list route-map FILTER in
```

Route Maps



- Unlike access control lists:
 - Route Maps can optionally specify a 'Set' action to change the packet or route information
 - Route Maps have many more possible options to match on a packet
 - You should include a sequence number when configuring Route Maps

```
route-map PBR permit 10
  match ip address ACL-A
  match length 0 1000 (0-1000 octets)
  set ip next-hop 10.10.10.1
```

```
interface GigabitEthernet 0/1
  ip policy route-map PBR
```

Route Maps



- Route Maps are a powerful tool to control routing policies
- They can be used to provide granular control for many router functions, such as:
 - Filtering routes
 - Setting the next hop based on the source of the packet (PBR Policy Based Routing)
 - Setting the metric for particular routes when redistributing
 - Controlling BGP path selection for particular routes

Route Map Syntax



```
Router(config)#route-map DEMO permit 10
Router(config-route-map)#match ?
  additional-paths  BGP Add-Path match policies
  as-path           Match BGP AS path list
  clns             CLNS information
  community        Match BGP community list
  extcommunity     Match BGP/VPN extended community list
  interface        Match first hop interface of route
  ip               IP specific information
  ipv6            IPv6 specific information
  length          Packet length
  local-preference Local preference for route
  mdt-group        Match routes corresponding to MDT group
  metric           Match metric of route
  mpls-label       Match routes which have MPLS labels
  policy-list      Match IP policy list
  route-type       Match route-type of route
  rpki            Match RPKI state of route
  security-group   Security Group
  source-protocol  Match source-protocol of route
  tag              Match tag of route
  track            tracking object
```

<https://t.me/learningnets>

Route Map Syntax



```
Router(config)#route-map DEMO permit 10
Router(config-route-map)#match ip address ACL-A
Router(config-route-map)#set ?
  aigp-metric      accumulated metric value
  as-path          Prepend string for a BGP AS-path attribute
  automatic-tag    Automatically compute TAG value
  clns             OSI summary address
  comm-list        set BGP community list (for deletion)
  community        BGP community attribute
  dampening        Set BGP route flap dampening parameters
  default          Set default information
  extcomm-list     Set BGP/VPN extended community list (for deletion)
  extcommunity     BGP extended community attribute
  global           Set to global routing table
  interface        Output interface
  ip               IP specific information
  ipv6             IPv6 specific information
  level            Where to import route
  lisp             Locator ID Separation Protocol specific information
  local-preference BGP local preference path attribute
  metric           Metric value for destination routing protocol
  metric-type      Type of metric for destination routing protocol
  mpls-label       Set MPLS label for prefix
  origin           BGP origin code
  tag              Tag value for destination routing protocol
  traffic-index    BGP traffic classification number for accounting
  vrf              Define VRF name
  weight           BGP weight for routing table
```



Route Map Syntax



- Each statement can match on multiple criteria
- Multiple match criteria in the same line is a logical 'OR'

```
route-map DEMO permit 10
  match ip address ACL-A ACL-B
route-map DEMO deny 20
  match ip address ACL-C
```

Route Map Syntax



- Multiple match criteria (of different types) under the same sequence number is a logical 'AND'
- The example below will match ACL-A and length 0-1000, or ACL-B and length 0-1000
- It will not match a packet which matches ACL-A and has a length of 1500 octets

```
route-map DEMO2 permit 10
  match ip address ACL-A ACL-B
  match length 0 1000
```

Route Map Syntax



- Multiple match criteria of the same type entered on different lines will be automatically converted to a logical 'OR'

```
route-map DEMO3 permit 10
  match ip address ACL-A
  match ip address ACL-B
```

is automatically converted to:

```
route-map DEMO3 permit 10
  match ip address ACL-A ACL-B
```

Route Map Syntax



- If a statement has no match criteria, it matches everything
- Sequence number 30 below permits everything not explicitly specified in earlier statements
- If it was not included then all other packets would be denied by the implicit deny at the end of the Route Map

```
route-map DEMO4 permit 10
  match ip address ACL-A ACL-B
route-map DEMO4 deny 20
  match ip address ACL-C
route-map DEMO4 permit 30
```

ACL/Prefix-List and Route Map Interaction



Route Map	Action
Permit	Allow the route into routing table
Deny	Do not allow the route into routing table
ACL / Prefix List	Action
Permit	Apply the Route Map action
Deny	Ignore the Route Map action and move on to the next Route Map statement

```
route-map DEMO5 deny 10
  match ip address 1
route-map DEMO5 permit 20
```

```
access-list 1 deny 192.168.1.0 0.0.0.255 ! Permitted by sequence 20
access-list 1 permit 192.168.0.0 0.0.255.255 ! Denied by sequence 10
```

- Because of the implicit Deny at the end, anything not explicitly specified in the ACL rolls through to the next Route Map statement

Standard ACLs for Route Filters – No Mask

- When a Standard ACL is used for route filtering in a Route Map:
- It works the same way as when used directly in a Distribute List
- If no subnet mask is configured, it will match on the **exact prefix** and **any valid subnet mask** (does not allow you to specify particular subnet mask)

- For example:

```
access-list 1 permit 10.17.32.0
```

- Will filter a route to 10.17.32.0 with any subnet mask of /19 or greater
- 10.17.32.0 in binary is **00001010.00010001.00100000.00000000**
- Will match 10.17.32.0/24 and 10.17.32.0/28 etc
- It will not match 10.17.32.1/32

Standard ACLs for Route Filters – No Mask

- Full configuration:

```
access-list 1 permit 10.17.32.0
!
route-map FILTER1 deny 10
  match ip address 1
route-map FILTER1 permit 20

router eigrp 100
  distribute-list route-map FILTER1 in
```

Comparison



- Full configuration when referenced directly in Distribute List:

```
access-list 1 deny 10.17.32.0
access-list 1 permit any
!
router ospf 1
  distribute-list 1 in
```

Standard ACLs for Route Filters – With Mask

- If a wildcard mask is configured, it will match prefixes which match **the range, with any valid subnet mask** (does not allow you to specify particular subnet mask)
- For example:

```
access-list 1 permit 10.17.32.0 0.0.0.255
```
- Will match any 10.17.32.x route with any mask of /19 or greater
- It will match 10.17.32.0/24 and 10.17.32.0/28 etc
- It will also match 10.17.32.1/32
- And 10.17.32.0/**19**

Standard ACLs for Route Filters – With Mask

- Full configuration:

```
access-list 2 permit 10.17.32.0 0.0.0.255
!
route-map FILTER2 deny 10
  match ip address 2
route-map FILTER2 permit 20
!
router eigrp 100
  distribute-list route-map FILTER2 in
```

Extended ACLs in Route Maps – Example 1

- If an extended ACL is used in a Route Map, then the source part of the ACL identifies the network address, and the destination part specifies the subnet mask (allowing you to specify a particular subnet mask)
- This example will filter out the route to 203.0.113.0/24

```
ip access-list extended FILTER_203.0.113.0/24
  permit ip host 203.0.113.0 host 255.255.255.0
```

```
route-map FILTER3 deny 10
  match ip address FILTER_203.0.113.0/24
route-map FILTER3 permit 20
```

```
router eigrp 100
  distribute-list route-map FILTER3 in
```

Extended ACLs in Route Maps – Example 2

- This example will filter out the route to any 203.0.x.x networks with exactly a /24 received on interface GigabitEthernet0/0

```
ip access-list extended FILTER_203.0.X.X/24
  permit ip 203.0.0.0 0.0.255.255 host 255.255.255.0
```

```
route-map FILTER4 deny 10
  match ip address FILTER_203.0.X.X/24
route-map FILTER4 permit 20
```

```
router eigrp 100
  distribute-list route-map FILTER4 in Gig0/0
```

Prefix Lists in Route Maps



- Prefix Lists work the same way in a Route Map as when referenced directly in a Distribute List

```
ip prefix-list PL1 permit 10.0.0.0/8 ge 18 le 20
```

```
route-map FILTER5 deny 10  
  match ip address prefix-list PL1  
route-map FILTER5 permit 20
```

```
router eigrp 100  
  distribute-list route-map FILTER5 in
```

Route Map Summary



- Use 'Permit' or 'Deny' in the Route-Map to specify the action, use 'Permit' in the ACL or Prefix-List
- If you specify multiple criteria of the same type under a single Route Map sequence number, a logical 'OR' is applied
- If you specify multiple criteria of a different type under a single Route Map sequence number, a logical 'AND' is applied
- If you need to specify specific subnet masks, use an Extended ACL in a Route Map, or a Prefix List

ACL, Prefix List, Distribute List, Route Map Summary

- When choosing whether to use an ACL, Prefix List or Route Map, you will usually be working from a configuration example which you can use as a template
- The simplest option for route filtering is typically a Standard ACL or a Prefix List, referenced directly in a Distribute List
- Route Maps are more commonly used for other features which require their if-then logic, such as PBR, granular redistribution, and BGP path selection

