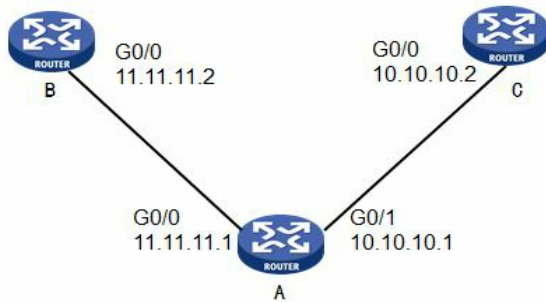


知 MSR56系列路由器OSPF路由过滤的案例

OSPF 李聪 2015-10-24 发表



如图所示，路由器B上面的Loop0地址为1.1.1.1/32，Loop1地址为2.2.2.2/32，G0/0地址为11.11.11.2/24；路由器C上面的Loop0地址为1.1.1.1/32，Loop1地址为2.2.2.2/32，G0/0地址为10.10.10.2/24；路由器A上面的G0/0地址为11.11.11.1/24，G0/1地址为10.10.10.1/24。现在在三台路由器上面配置了ospf，都在area 0区域中。现在要求在A上面配置路由过滤，访问1.1.1.1的时候去往B，访问2.2.2.2的时候去往C。

未配置路由过滤的时候在A上面看到的路由信息：

```
[H3C]
[H3C]dis ip routing-table
Destinations : 18      Routes : 20

Destination/Mask    Proto   Pre  Cost           NextHop         Interface
0.0.0.0/32          Direct  0    0              127.0.0.1       InLoop0
1.1.1.1/32          O_INTRA 10   1              10.10.10.2      GEO/1
                    O_INTRA 10   1              11.11.11.2      GEO/0
2.2.2.2/32          O_INTRA 10   1              10.10.10.2      GEO/1
                    O_INTRA 10   1              11.11.11.2      GEO/0
10.10.10.0/24       Direct  0    0              10.10.10.1      GEO/1
10.10.10.0/32       Direct  0    0              10.10.10.1      GEO/1
10.10.10.1/32       Direct  0    0              127.0.0.1       InLoop0
10.10.10.255/32     Direct  0    0              10.10.10.1      GEO/1
11.11.11.0/24       Direct  0    0              11.11.11.1      GEO/0
11.11.11.0/32       Direct  0    0              11.11.11.1      GEO/0
11.11.11.1/32       Direct  0    0              127.0.0.1       InLoop0
11.11.11.255/32     Direct  0    0              11.11.11.1      GEO/0
127.0.0.0/8         Direct  0    0              127.0.0.1       InLoop0
127.0.0.0/32        Direct  0    0              127.0.0.1       InLoop0
127.0.0.1/32        Direct  0    0              127.0.0.1       InLoop0
127.255.255.255/32 Direct  0    0              127.0.0.1       InLoop0
224.0.0.0/4         Direct  0    0              0.0.0.0         NULL0
224.0.0.0/24        Direct  0    0              0.0.0.0         NULL0
255.255.255.255/32 Direct  0    0              127.0.0.1       InLoop0
```

可见，在A上面能够看到等价路由。

- 1.需要针对目的地址以及下一跳做过滤。
- 2.使用filter-policy { acl-number| ip-prefix ip-prefix-name [gateway ip-prefix-name]}方法。一个ospf进程里面只可以配置一条，而且即使在地址前缀里面将目的地址和下一跳加进去，可是地址之间的关系是“与”的关系，要么全部禁止要么只能访问一台路由器。此方法行不通。
- 3.使用route-policy的方法，route-policy之间的节点是“或”的关系，if-match是“与”的关系，可以实现需求，但是在ospf进程引入策略的时候提示引入的ospf路由不能在同一个进程里面使用。因此也不行。
- 4.filter-policy route-policy route-policy-name import方法，这个方法刚好结合上述两种方法的优点，因此本案例只能通过这个过滤方式来实现。

- 1.ip prefix-list abc index 10 permit 1.1.1.1 32 // 匹配1.1.1.1/32的路由
- 2.ip prefix-list abcd index 10 permit 2.2.2.2 32 //匹配2.2.2.2/32的路由
- 3.ip prefix-list next index 10 permit 11.11.11.2 32 //匹配下一跳为11.11.11.2
- 4.ip prefix-list next1 index 10 permit 10.10.10.2 32 //匹配下一跳为10.10.10.2

5.route-policy test deny node 10

if-match ip address prefix-list abc

if-match ip next-hop prefix-list next1 //节点10为deny, 匹配了去往1.1.1.1/32的下一跳为10.10.10.2, 然后过滤掉这条路由。

6.route-policy test deny node 20

if-match ip address prefix-list abcd

if-match ip next-hop prefix-list next //节点20为deny, 匹配了去往2.2.2.2/32的下一跳为11.11.11.2, 然后过滤掉这条路由。

7.route-policy test permit node 30 //空节点, 放开所有路由。通常在多个deny节点后设置一个不含if-match子句和apply子句的permit节点, 用于允许其它的路由通过。

Node10和node20是或的关系, 因此deny的路由都被过滤掉了, 但是没有放开其他的路由, 因此最后需要放开其他全部的路由, 使用node30来实现。

8.ospf 1

filter-policy route-policy test import //在ospf里面引用filter-policy route-policy

最后路由成功过滤:

```
[H3C]dis ip routing-table
Destinations : 18          Routes : 18

Destination/Mask    Proto   Pre  Cost           NextHop           Interface
0.0.0.0/32          Direct  0    0              127.0.0.1         InLoop0
1.1.1.1/32          O_INTRA 10   1              11.11.11.2        GEO/0
2.2.2.2/32          O_INTRA 10   1              10.10.10.2        GEO/1
10.10.10.0/24       Direct  0    0              10.10.10.1        GEO/1
10.10.10.0/32       Direct  0    0              10.10.10.1        GEO/1
10.10.10.1/32       Direct  0    0              127.0.0.1         InLoop0
10.10.10.255/32     Direct  0    0              10.10.10.1        GEO/1
11.11.11.0/24       Direct  0    0              11.11.11.1        GEO/0
11.11.11.0/32       Direct  0    0              11.11.11.1        GEO/0
11.11.11.1/32       Direct  0    0              127.0.0.1         InLoop0
11.11.11.255/32     Direct  0    0              11.11.11.1        GEO/0
127.0.0.0/8         Direct  0    0              127.0.0.1         InLoop0
127.0.0.0/32        Direct  0    0              127.0.0.1         InLoop0
127.0.0.1/32        Direct  0    0              127.0.0.1         InLoop0
127.255.255.255/32 Direct  0    0              127.0.0.1         InLoop0
224.0.0.0/4         Direct  0    0              0.0.0.0           NULL0
224.0.0.0/24        Direct  0    0              0.0.0.0           NULL0
255.255.255.255/32 Direct  0    0              127.0.0.1         InLoop0
```

使用filter-policy { acl-number| ip-prefix ip-prefix-name [gateway ip-prefix-name] }方法。一个ospf进程里面只能配置一条, 使用route-policy的方法, route-policy之间的节点是“或”的关系, if-match是“与”的关系, 可以实现需求, 但是在ospf进程引入策略的时候提示引入的ospf路由不能在同一个进程里面使用。filter-policy route-policy route-policy-name import方法, 这个方法刚好结合上述两种方法的优点, 因此本案例只能通过这个过滤方式来实现。