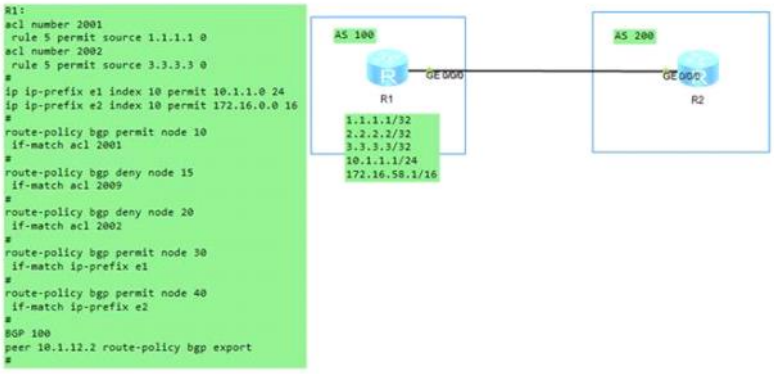
IP-Prefix和ACL区别

2021年11月17日 20:17



注释：原题中可能没有node 15，ACL 2009也没有创建，如果遇到当前的情况如何处理?
匹配的ACL 2009时，发现该ACL不存在，默认代表所有
则按当前空node的动作处理，假设是deny 拒绝15之后的所有路由，permit的话是允许所有。

1、R1通过network方式注入5条路由，对邻居发布路由时调用路由策略，请问在R2上能看到几条路由，写出掩码。（2分）
2、如果路由策略中配置了router route-policy bgp permit node 50 后，在R2上能看到几条路由，写出掩码。(2分)
3、写出ACL与前缀列表的区别。（6分）

1，R1通过network方式注入5条路由，对邻居发布路由时调用路由策略，请问在R2上能看到几条路由，写出掩码。（2分）
答:
R2上display bgp routing-table 仅看到1.1.1.1/32 。
分析:

```
acl number 2001
rule 5 permit source 1.1.1.1 0
acl number 2002
rule 5 permit source 3.3.3.3 0
#
ip ip-prefix e1 index 10 permit 10.1.1.0 24
ip ip-prefix e2 index 10 permit 172.16.0.0 16
#
route-policy bgp permit node 10    //放行1.1.1.1
if-match acl 2001
#
route-policy bgp deny node 15  //删除其他所有    //--如果此处是删除所有，那么后面的都不执行
                                                  //--如果此处是permit node 15，那么放行所有，后面也都不执行
if-match acl 2009
#
route-policy bgp deny node 20  //拒绝 3.3.3.3
if-match acl 2002
#
route-policy bgp permit node 30  //放行10.1.1.0/24
if-match ip-prefix e1
#
route-policy bgp permit node 40  //放行172.16.0.0/16
if-match ip-prefix e2
#
```

变种：如果题中，没有上述node 15 选项
解答：R2上display bgp routing-table 仅看到1.1.1.1/32 、10.1.1.0/24 、172.16.0.0/16

2，哪些路由被过滤掉(2分)
疑惑？？？？这个和上一题，不是一个意思？？？？送分？？？？？
既然正经问了，就正经答

解答：根据考场情况
情况一：有deny node 15
因为node 15 匹配的是空列表，默认代表所有，所以node15代表过滤所有路由。
node 10优先执行，放行了1.1.1.1/32 。
所以过滤了除了1.1.1.1/32的所有路由 ，即 2.2.2.2/32、3.3.3.3/32、10.1.1.0/24、172.16.0.0/16 。

情况二：有permit node 15
因为node 15 匹配的是空列表，默认代表所有，所以node15代表放行所有路由。
即没有过滤任何路由。

情况三：没有node 15
node 20中过滤了3.3.3.3/32
2.2.2.2/32没有被node命中，默认被拒绝过滤掉
所以过滤了3.3.3.3/32、2.2.2.2/32

3，IP-prefix和ACL的区别 （6分）
解答：
前缀列表和ACL都可以用于路由控制，作为匹配路由的工具，同时ACL还可以匹配流量。他们的区别主要分为一下几点阐述：

1、使用范围：
ACL的使用范围更广泛，如下：
标准ACL，针对source 可以匹配路由和数据
扩展ACL，可针对具体流量，可以匹配指定源到目标及协议优先级等具体参数
二层ACL，可以匹配二层流量，针对特定802.1P、vlan id、源目mac地址、二层协议类型进行精确匹配。
IPv6 ACL，可以针对IPv6环境，实现同上述德标准和扩展ACL功能。
而前缀列表只能针对ipv4、ipv6的网络前缀进行路由层面的控制。

2、路由匹配的灵活性
ACL，使用通配符，更便于匹配有规律性但不连续的网络。
例如：192.168.x.0/24的偶数路由可以配置如下ACL：

```
 acl 2000
  rule permit source 192.168.0.0  0.0.254.0
 #
```

而前缀列表匹配的网络范围，其网络号必须连续，无法实现上述效果。

3、路由的精确性
ACL的通配符关心网络号，但是掩码不好精确匹配，所以可以使用前缀列表更加精确匹配路由的网络前缀范围以及掩码长度范围。
例如：
192.168.0.0/16
192.168.0.0/22
192.168.0.0/24

使用acl：

```
 acl 2000
  rule permit source 192.168.0.0  0
```

#
匹配的为网络号为192.168.0.0的路由，不关心掩码长度，所以此时3条路由都配匹配上，无法区分。

使用ip-prefix：

ip ip-prefix test permit 192.168.0.0 16 greater-equal 20 less-equal 24

匹配的网络号为192.168.0.0，掩码长度范围在20~24之间的路由，所以可以匹配192.168.0.0/22和192.168.0.0/24。

ip ip-prefix test permit 192.168.0.0 16

不指定掩码长度范围，那么掩码长度就等于前缀长度，此时只匹配192.168.0.0/16这一条路由。

4、未命中规则的处理方式

ACL中不管是“通过”，还是“拒绝”，最终实际的动作是由应用ACL的业务模块决定的。不同业务模块对未命中或者命中的报文处理方式不同。

在匹配路由时，ACL中配置规则但未命中，默认动作为deny，拒绝其他路由通过

例如：

192.168.1.0/24
192.168.2.0/24
#
acl 2000
 rule 5 permit source 192.168.1.0 0
isis 1
 filter-policy 2000 export

使用上面规则时，192.168.2.0/24路由会拒绝向其他设备通告

但在匹配流量时，流策略中的ACL默认动作是permit，如果ACL中存在规则但报文未命中规则，该报文仍可以正常通过。

而Telnet，Stelnet，SNMP等中的ACL默认动作是deny，如果ACL中存在规则但报文未命中规则，该报文会被拒绝通过。

黑名单模块中的ACL处理机制与其他模块有所不同。无论ACL规则是permit还是deny，只要命中规则，该报文都会被丢弃

而IP-Prefix用于匹配路由时，具有隐藏deny动作，存在规则但未命中默认拒绝