访问控制表的工作原理和详解

1.访问控制列表(ACL)的工作原理

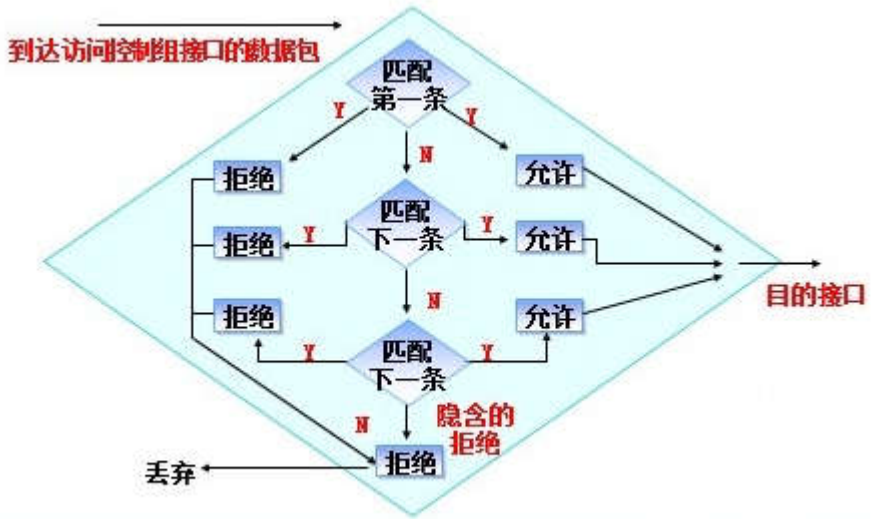
ACL 使用包过滤技术，在路由器上读取 OSI 七层模型的第 3 层和第 4 层包头中的信息。如源地址，目标地址，源端口，目标端口等，根据预先定义好的规则，对包进行过滤，从而达到访问控制的目的。

ACl 是一组规则的集合，它应用在路由器的某个接口上。对路由器接口而言，访问控制列表有两个方向。

出：已经过路由器的处理，正离开路由器的数据包。

入：已到达路由器接口的数据包。将被路由器处理。

如果对路由器的某接口应用了 ACL，那么路由器对数据包应用该组规则进行顺序匹配，使用匹配即停止的，不匹配则使用默认规则的方式来过滤数据包。如下图：



2.访问控制列表的类型

2.1 标准访问控制列表：

根据数据包的源 IP 地址来允许或拒绝数据包，标准访问控制列表的访问控制列表号是 1-99。

2.2 扩展访问控制列表：

根据数据包的源 IP 地址，目的 IP 地址，指定协议，端口和标志，来允许或拒绝数据包。扩展访问控制列表的访问控制列表号是 100-199

3.配置访问控制表

3.1 配置标准控制列表

创建标准 ACL 的语法如下：

```
    Router(config)#access-list access-list-number {permit|
deny} source [souce-wildcard]
```

下面是命令参数的详细说明

access-list-number：访问控制列表号，标准 ACL 取值是 1-99。

permit|deny：如果满足规则，则允许/拒绝通过。

source：数据包的源地址，可以是主机地址，也可以是网络地址。

source-wildcard：通配符掩码，也叫做反码，即子网掩码去反值。如：正常子网掩码 255.255.255.0 取反则是 0.0.0.255。

删除已建立的标准 ACL 语法如下：

```
    Router(config)#no access-list access-list-number
```

例如：创建一个 ACL 允许 192.168.1.0 网段的所有主机。

```
    Router(config)#access-list 1 permit 192.168.1.0 0.0.0.2
55
```

例如：创建一个 ACL 允许某个主机。

```
    Router(config)#access-list 1 permit host 10.0.0.1
```

例如：创建一个默认 ACL 拒绝所有主机访问。

```
    Router(config)#access-list 1 deny any
```

注意：上述中的关键字 host 可以指定一个主机地址，而不用写子网反码，而 any 可以代表所有主机。

3.2 配置扩展访问控制列表

创建扩展的 ACL 语法如下：

```
    Router(config)#access-list access-list-number {permit|
deny} protocol {source souce-wildcard destination destinat
ion-wildcard} [operator operan]
```

下面是命令参数的详细说明

access-list-number：访问控制列表号，扩展 ACL 取值是 100-199。

permit|deny：如果满足规则，则允许/拒绝通过。

protocol：用来指定协议的类型，如 IP，TCP，UDP，ICMP 等。

source、destination：源和目的，分别用来标示源地址和目的地址。

souce-wildcard、destination-wildcard：子网反码，souce-wildcard 是源反码，destination-wildcard 是目标反码。

operator operan：lt(小于)、gt(大于)、eq(等于)、neq(不等于)一个端口号。

删除已建立的扩展 ACL 语法如下：

```
Router(config)#no access-list access-list-number
```

例如：允许 192.168.1.0/24 访问 192.168.2.0/24，而拒绝其他所有主机访问。

```
Router(config)#access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

Router(config)#access-list 101 deny ip any any
```

例如：拒绝网络 192.168.1.0/24 访问 FTP 服务器 192.168.2.100/24，而允许其他主机访问。

```
Router(config)#access-list 102 deny tcp 192.168.1.0 0.0.0.255 host 192.168.2.100 eq 21

Router(config)#access-list 102 permit ip any any
```

例如：禁止网络 192.168.1.0/24 中的主机 ping 同服务器 192.168.2.200/24，而允许其它主机访问。

```
Router(config)#access-list 103 deny icmp 192.168.1.0 0.0.0.255 host 192.168.1.200 echo

Router(config)#access-list 103 permit ip any any
```

将创建好的 ACL 应用与路由器的接口上

不管是标准 ACL 还是扩展 ACL 只有将创建好的 ACL 应用与路由器的接口上才算是有效的。语法如下：

```
    Router(config-if)#ip access-group access-list-number
{in|out}
```

参数解释如下：

access-list-number：创建 ACL 时指定的访问控制列表号

in：应用到入站接口。

out：应用出站接口。

取消接口上的 ACL 应用可以使用如下命令：

Router(config-if)#no ip access-group access-list-number {in|out}

可以使用 show access-lists 命令查看 ACL 配置。

注意：不管是标准 ACL 或者是扩展 ACL，只要应用了该规则就不可以在向里面添加新的规则了，只能是删除整个 ACL。这样很不方便我们管理 ACL，那么我们改这么办呢？下面我们来讲解命名访问控制列表。

3.3 配置命名访问控制列表

所谓的命名控制列表就是给控制列表取个名字，而不是想上面所述的使用访问控制列表号。我们通过命令访问控制列表可以很方便的管理 ACL 规则，可以随便添加和删除规则，而无需删除整个访问控制列表了。

创建命名访问控制列表的语法如下：

```
    Router(config)#ip access-list {standard|extended} acce
ss-list-name
```

下面是命令参数的详细说明

standard:创建标准的命名访问控制列表。

extended:创建扩展的命名访问控制列表。

access-list-name:命名控制列表的名字，可以是任意字母和数字的组合。

标准命名 ACL 语法如下：

```
    Router(config-std-nacl)#[Sequence-Number] {permit|den
y} source [souce-wildcard]
```

扩展命名 ACL 语法如下：

```
    Router(config-ext-nacl)#[Sequence-Number] {permit|den
y} protocol {source souce-wildcard destination destination
-wildcard} [operator operan]
```

无论是配置标准命名 ACL 语句还是配置扩展命名 ACL 语句，都有一个可选参数 Sequence-Number。Sequence-Number 参数表明了配置的 ACL 语句在命令 ACL 中所处的位置，默认情况下，第一条为 10，第二条为 20，以此类推。Sequence-Number 可以很方便地将新添加的 ACL 语句插于到原有的 ACL 列表的指定位置，如果不选择 Sequence-Number，默认添加到 ACL 列表末尾并且序列号加 10。

删去已创建的命名 ACL 语法如下：

```
    Router(config)#no ip access-list {standard|extended} ac
cess-list-name
```

对于命名 ACL 来说，可以删除单条 ACL 语句，而不比删除整个 ACL。并且 ACL 语句可以有选择的插入到列表中的某个位置，使得 ACL 配置更加方便灵活。

如果要删除某一 ACL 语句，可以使用“no Sequence-Number”或“no ACL”语句

两种方式。

例如：将一条新添加的 ACL 加入到原有标准命名 ACL 的序列 15 的位置。内容为允许主机 192.168.1.1/24 访问 Internet。

```
    Router(config)#ip access-list standard test1

    Router(config-std-nacl)#15 permit host 192.168.1.1
```

例如：创建扩展命名 ACL，内容为拒绝 192.168.1.0/24 访问 FTP 服务器 192.168.2.200/24，允许其他主机。

```
    Router(config)#ip access-list extended test2

    Router(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host 192.168.2.200 eq 21

    Router(config-ext-nacl)#permit ip any any
```

将命名 ACL 应用于接口语法如下：

```
    Router(config-if)#ip access-group aaccess-list-name {in|out}
```

取消命名 ACL 的应用语法如下：

```
    Router(config-if)#no ip access-group aaccess-list-name {in|out}
```