ACL 访问控制列表

ACL 是为了对某些数据流进行过滤，达到实现基本网络安全的目的的手段之一。

ACL 大概可以分为标准，扩展以及命名 ACL

1．标准 ACL

标准 ACL 最简单，是通过使用 IP 包中的源 IP 地址进行过滤，表号范围 1-99 或 1300-1999；

2．扩展 ACL

扩展 ACL 比标准 ACL 具有更多的匹配项，功能更加强大和细化，可以针对包括协议类型、源地址、目的地址、源端口、目的端口、TCP 连接建立等进行过滤，表号范围 100-199 或 2000-2699；

3．命名 ACL

以列表名称代替列表编号来定义 ACL，同样包括标准和扩展两种列表。

在访问控制列表的学习中，要特别注意以下两个术语。

1．通配符掩码：一个 32 比特位的数字字符串,它规定了当一个 IP 地址与其他的 IP 地址进行比较时，该 IP 地址中哪些位应该被忽略。通配符掩码中的“1”表示忽略 IP 地址中对应的位，而“0”则表示该

位必须匹配。两种特殊的通配符掩码是“255.255.255.255”和“0.0.0.0”，前者等价于关键字“any”，而后者等价于关键字“host”；

2．Inbound 和 outbound：当在接口上应用访问控制列表时，用户要指明访问控制列表是应用于流入数据还是流出数据。

总之，ACL 的应用非常广泛，它可以实现如下的功能：

1．拒绝或允许流入（或流出）的数据流通过特定的接口；

2．为 DDR 应用定义感兴趣的数据流；

3．过滤路由更新的内容；

4．控制对虚拟终端的访问；

5．提供流量控制。

标准 ACL 配置命令：

Router(config)#access-list access-list_number permit/deny 源网段 反掩码

扩展 ACL 配置命令：

Router(config)#access-list access-list_number permit/deny icmp/tcp/udp/ospf/eigrp 源网段 反掩码 目的网段 反掩码 ？（问号之后可根据需要选择）

命名 ACL 配置命令：

Router(config)#ip access-list standard stand

//对标准命名ACL取名为stand

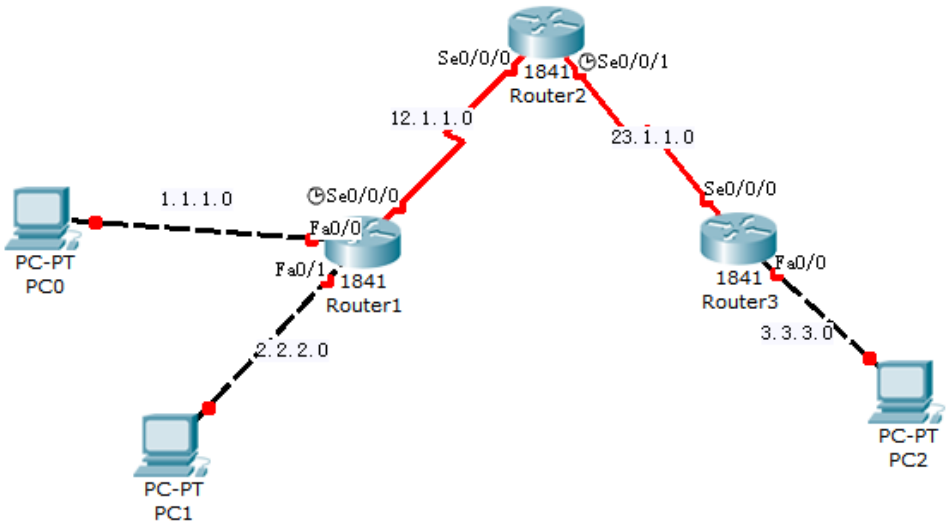
Router(config)#ip access-list extended ext1

//对扩展命名ACL取名为ext1

之后在接口下进行应用，详细情况可见下列例子。

实验五 标准访问控制列表 ACL

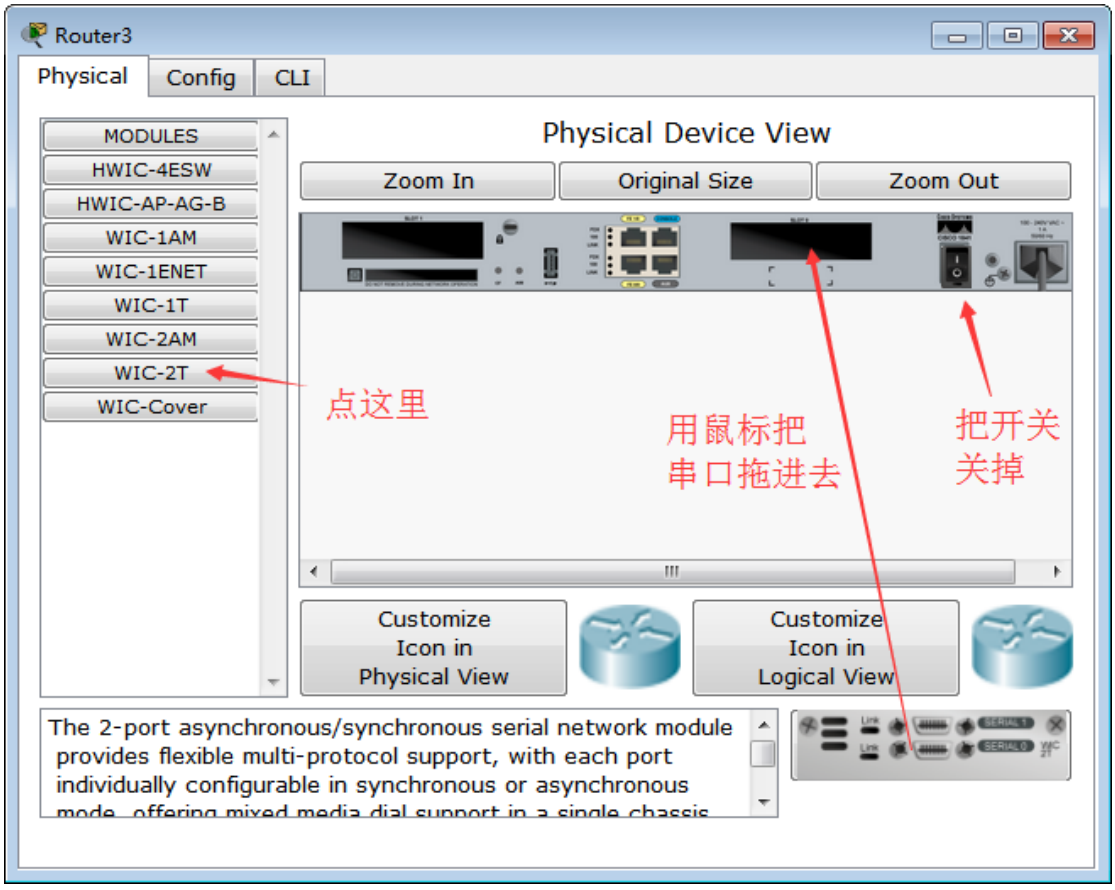
1.拓扑结构

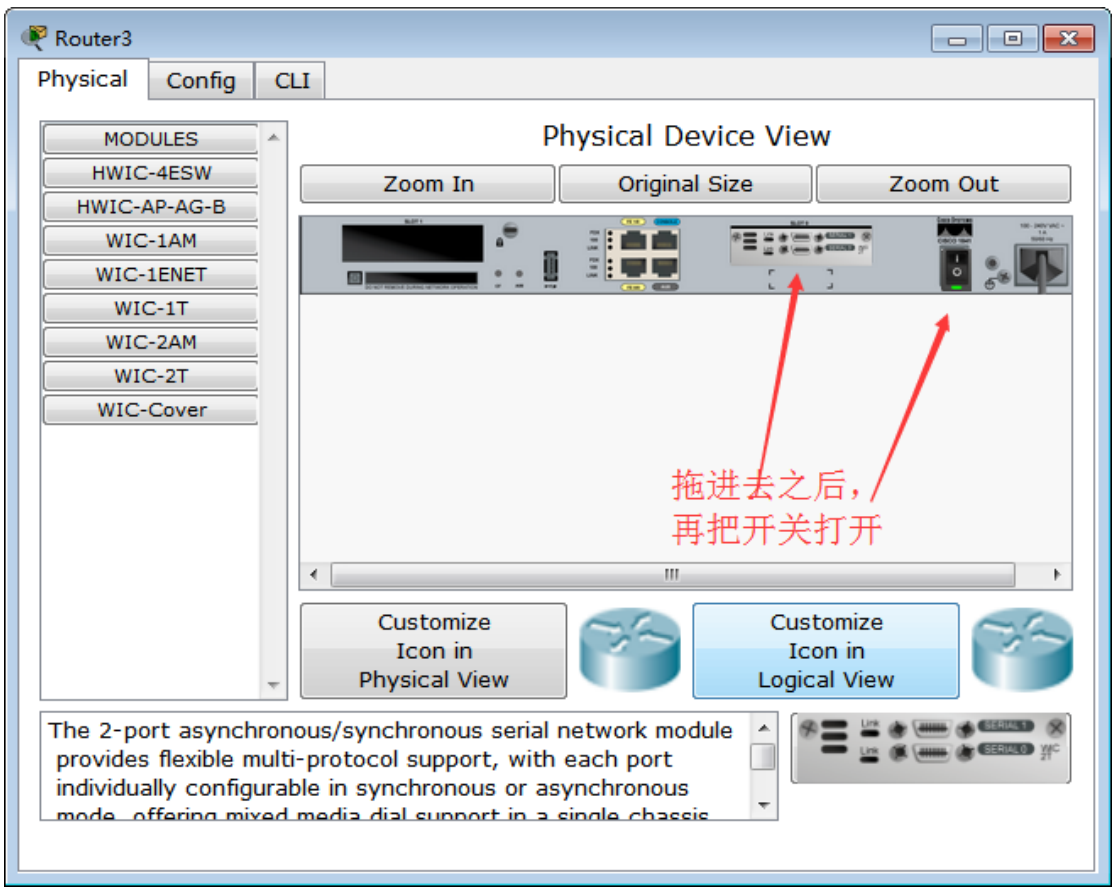


标准 ACL 拓扑图

本实验希望 PC1 所在网段无法访问路由器 R2（PC1 ping 不通，PC0 和 PC2 可以 ping 通），而只允许主机 PC2 访问路由器 R2 的 telnet 服务（PC2 能 telnet，PC0 和 PC2 不能 telnet）。

默认情况下，1841 型号的路由器不带 serial 口（串口），需要手动添加，添加方式如下：


Router3
Physical
Config
CLI
MODULES
HWIC-4ESW
HWIC-AP-AG-B
WIC-1AM
WIC-1ENET
WIC-1T
WIC-2AM
WIC-2T
WIC-Cover
Physical Device View
Zoom In
Original Size
Zoom Out
点这里
用鼠标把
串口拖进去
把开关
关掉
Customize
Icon in
Physical View
Customize
Icon in
Logical View
The 2-port asynchronous/synchronous serial network module
provides flexible multi-protocol support, with each port
individually configurable in synchronous or asynchronous


Router3
Physical
Config
CLI
MODULES
HWIC-4ESW
HWIC-AP-AG-B
WIC-1AM
WIC-1ENET
WIC-1T
WIC-2AM
WIC-2T
WIC-Cover
Physical Device View
Zoom In
Original Size
Zoom Out
拖进去之后，
再把开关打开
Customize
Icon in
Physical View
Customize
Icon in
Logical View
The 2-port asynchronous/synchronous serial network module
provides flexible multi-protocol support, with each port
individually configurable in synchronous or asynchronous

另，各路由器的端口 IP 如下：

路由器	端口 Fa0/0	端口 Fa0/1	端口 Se0/0/0	端口 Se0/0/1
R1	1.1.1.2	2.2.2.2	12.1.1.1	
R2			12.1.1.2	23.1.1.2
R3	3.3.3.3		23.1.1.3	

根据上面的 IP 信息分别对 R1，R2，R3 配置 IP 地址，

例如：对 R1 的配置如下：

```
R1(config)#int f0/0
R1(config-if)#ip add 1.1.1.2 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#int f0/1
R1(config-if)#ip add 2.2.2.2 255.255.255.0
R1(config-if)#no shutdown
R1(config)#int serial 0/0/0
R1(config-if)#ip add 12.1.1.1 255.255.255.0
R1(config-if)#clock rate 9600 // 给串口设定传输速率
R1(config-if)#no shutdown
```

同理，对 R2，R3 进行配置。配置完路由 IP 地址后，还要将 PC 机的 IP 地址配好，PC 的 IP 根据需要自行设计，默认网关即为直连路由的接口 IP 地址。

对三个路由器做动态路由协议配置，本实验使用 OSPF 协议。做完配

置后，可以实现 PC 和路由间的全拼通。之后做 ACL，用来实现某些限制条件。

为了实验方便，实现完 PC 和路由器之间的全拼通之后，建议另存一份，方便后面的实验。

为了达到 PC1 所在网段无法访问（ping 不通）路由器 R2，而只允许主机 PC2 访问路由器 R2 的 telnet 服务的目的，主要是在 R2 上做配置。

下面演示如何在 R2 的端口应用 ACL 规则

```
//定义 ACL 规则 10 为拦截所有关于 2.2.2.0 网端的数据，进和出的数据
R2(config)#access-list 10 deny 2.2.2.0 0.0.0.255
R2(config)#int s0/0/0
R2(config-if)#ip access-group 10 in    //在接口下应用
```

此时，发现不单单是 PC1 ping 不通 R2，连 PC0 也无法 ping 通 R2。同时，PC2 也无法 ping 通 PC1 或者 PC0 了。所以配置了 ACL 之后，R2 到 R1 之间的所有数据通讯都被截断了。这时，我们需要添加一些过滤规则。

```
R2(config)#access-list 10 permit any //允许除了明确 deny 以外的所有数据
R2(config)#int s0/0/0
R2(config-if)#ip access-group 10 in    //在接口下应用
```

这样，我们就实现了，只有 PC1 不能 ping 通 R2 的要求。大家

可以再测试一下 PC2 和 PC0 的连通性。接下来，配置只允许 PC2 能访问 R2 的 telnet 服务，其他 PC 不能访问。

R2(config)#access-list 20 permit host 3.3.3.1 //指定允许某个具体的 IP 地址，这里 3.3.3.1 是 PC2 的 IP

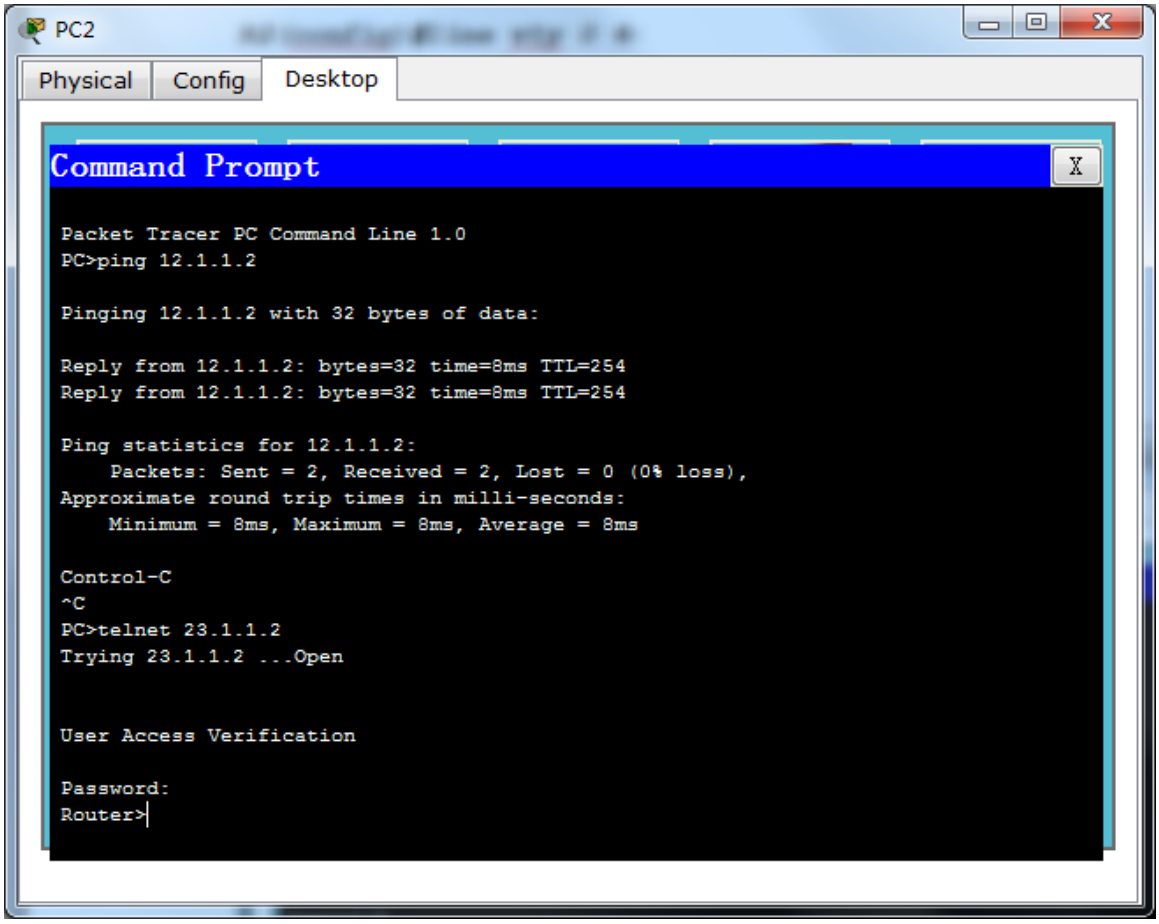
R2(config)#int s0/0/1 //在接口下应用远程登录

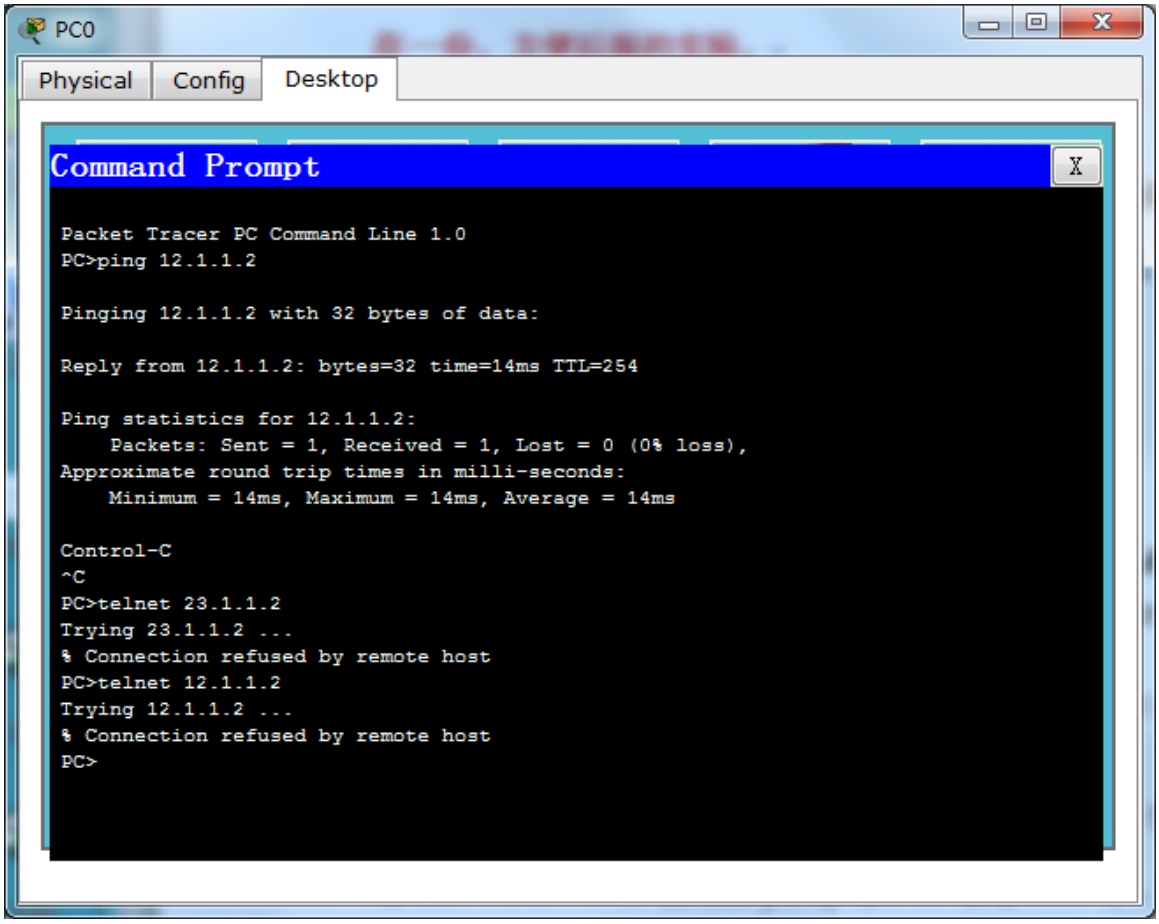
```
R2(config)#line vty 0 4
R2(config-line)#access-class 20 in
R2(config-line)#pass 123456
R2(config-line)#login
R2(config-line)#end
```

access-class 与 access-group 的区别

当你配置好 ACL 之后，想把 ACL 应用到某个接口下，如 Ethernet，FastEthernet 等物理接口下，那么需要使用 access-group，如果想要把 ACL 应用到 VTY 的接口下用于对网络设备的访问控制，那么需要用 access-class。

然后测试 PC2 和 PC0 的连通性以及是否能 telnet 到 R2 上。


PC2
Physical
Config
Desktop
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 12.1.1.2
Pinging 12.1.1.2 with 32 bytes of data:
Reply from 12.1.1.2: bytes=32 time=8ms TTL=254
Reply from 12.1.1.2: bytes=32 time=8ms TTL=254
Ping statistics for 12.1.1.2:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 8ms, Maximum = 8ms, Average = 8ms
Control-C
^C
PC>telnet 23.1.1.2
Trying 23.1.1.2 ...Open
User Access Verification
Password:
Router>


PC0
Physical
Config
Desktop
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 12.1.1.2
Pinging 12.1.1.2 with 32 bytes of data:
Reply from 12.1.1.2: bytes=32 time=14ms TTL=254
Ping statistics for 12.1.1.2:
Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 14ms, Maximum = 14ms, Average = 14ms
Control-C
^C
PC>telnet 23.1.1.2
Trying 23.1.1.2 ...
% Connection refused by remote host
PC>telnet 12.1.1.2
Trying 12.1.1.2 ...
% Connection refused by remote host
PC>

扩展 ACL

本实验要求 PC0 既不能 ping 通 R2，又不能 telnet R2，PC1 只能 telnet R2，不能 ping 通 R2，PC2 只能 telnet R2 不能 ping 通 R2。

删除实验 1 中定义的 ACL，保留 OSPF 的配置。

R2：

```
R2>en
R2#conf t
R2(config)#no access-list 10
R2(config)#no access-list 20
```

其余配置不变。

先配置 R2 上的 telnet 服务

```
R2(config)#line vty 0 4
R2(config-line)#pass 123456
R2(config-line)#login
```

这时，所有的 PC 都能连接 R2 的 telnet 服务。

开始设置 R2 的 ACL 规则

R2：

```
R2(config)#access-list 100 deny ip 1.1.1.0 0.0.0.255 host 12.1.1.2
R2(config)#access-list 100 deny ip 1.1.1.0 0.0.0.255 host 23.1.1.2

R2(config)#access-list 100 deny icmp 2.2.2.0 0.0.0.255 host 12.1.1.2
```

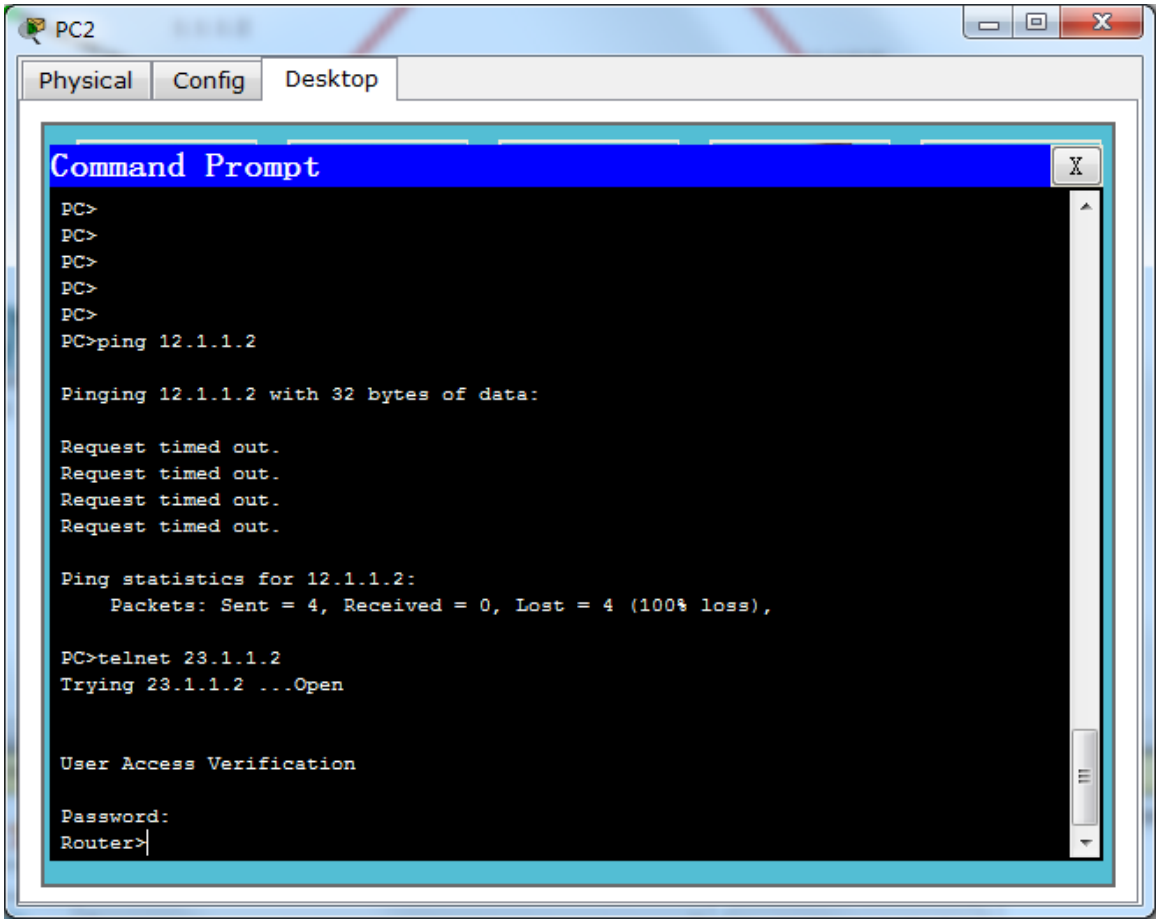
```
R2(config)#access-list 100 deny icmp 2.2.2.0 0.0.0.255 host
23.1.1.2

R2(config)#access-list 100 permit ip any any
R2(config)#int s0/0/0
Router(config-if)#ip access-group 100 in
```

这时，PC0 已经不能 ping 通 R2 以及连接 R2 的 telnet 服务。PC1 能连上 R2 的 telnet 服务，却不能 ping 通。接着配置 R3 的 ACL。

R3：

```
Router(config)#access-list 101 deny icmp 3.3.3.0  0.0.0.255
host 23.1.1.2
Router(config)#access-list 101 deny icmp 3.3.3.0  0.0.0.255
host 12.1.1.2
Router(config)#access-list 101 permit ip any any
Router(config)#int f0/0
Router(config-if)#ip access-group 101 in
```


PC2
Physical
Config
Desktop
Command Prompt
PC>
PC>
PC>
PC>
PC>
PC>ping 12.1.1.2
Pinging 12.1.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 12.1.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>telnet 23.1.1.2
Trying 23.1.1.2 ...Open
User Access Verification
Password:
Router>

命名 ACL

命名 ACL 允许在标准 ACL 和扩展 ACL 中， 使用字符串代替前面所使用的数字来表示 ACL。

命名 ACL 还可以被用来从某一特定的 ACL 中删除个别的控制条目， 这样可以让网络管理员方便地修改 ACL。

本次实验中，我们用命名 ACL 实现前面的两个要求

1）我们在 R2 上配置命名的标准 ACL 来实现第一个实验要求,即 PC1 无法 ping 通 R2，PC0 可以 ping 通 R2，只允许 PC2 访问路由器 R2 的 telnet 服务。

R2:

Router(config)#ip access-list standard stand //对标准命名 ACL 取名为 stand

Router(config-std-nacl)#deny 2.2.2.0 0.0.0.255
Router(config-std-nacl)#permit any
Router(config-std-nacl)#exit
Router(config)#int s0/0/0
Router(config-if)#ip access-group stand in
Router(config-if)#exit

Router(config)#ip access-list standard class //对标准命名 ACL 取名为 class

Router(config-std-nacl)#permit host 3.3.3.x(PC2 的 IP)

Router(config-std-nacl)#exit
Router(config)#int s0/0/1
Router(config-if)#line vty 0 4
Router(config-line)#pass 123456
Router(config-line)#access-class class in

Router(config-line)#login

2）下面用命名 ACL 完成扩展 ACL 方面的实验操作，完成实验 2 的条件，即 PC0 既不能 ping 通 R2，又不能 telnet R2，而 PC1/PC2 只能 telnet R2 不能 ping 通 R2。

首先我们先删除掉 R2 中的命名 ACL 配置，

R2:

```
Router#conf t
Router(config)#no ip access-list standard stand
Router(config)#no ip access-list standard class
Router#show access-lists
Standard IP access list class
```

删除成功

先配置 R2 上的 telnet 服务

```
R2(config)#line vty 0 4
R2(config-line)#pass 123456
R2(config-line)#login
```

之后对 R2，R3 做命名 ACL 配置，

R2:

```
Router#conf t
Router(config)#ip access-list extended ext1
Router(config-ext-nacl)#deny ip 1.1.1.0 0.0.0.255 host
12.1.1.2
Router(config-ext-nacl)#deny ip 1.1.1.0 0.0.0.255 host
```

```
23.1.1.2
Router(config-ext-nacl)#deny icmp 2.2.2.0 0.0.0.255 host
12.1.1.2
Router(config-ext-nacl)#deny icmp 2.2.2.0 0.0.0.255 host
23.1.1.2
Router(config-ext-nacl)#permit ip any any
Router(config-ext-nacl)#exit
Router(config)#int s0/0/0
Router(config-if)#ip access-group ext1 in
Router(config-if)#exit
```

当用 show access-list 命令事会显示和扩展 ACL 中一样的条件。

```
Router#sh access-list
Extended IP access list ext1
    10 permit tcp 2.2.2.0 0.0.0.255 host 12.1.1.2 eq telnet
    20 permit tcp 2.2.2.0 0.0.0.255 host 23.1.1.2 eq telnet
```

R3:

```
Router#conf t
Router(config)#ip access-list extended ext2
Router(config-ext-nacl)#deny icmp 3.3.3.0 0.0.0.255 host
23.1.1.2
Router(config-ext-nacl)#deny icmp 3.3.3.0 0.0.0.255 host
12.1.1.2
Router(config-ext-nacl)#permit ip any any
Router(config-ext-nacl)#exit
Router(config)#int f0/0
Router(config-if)#ip access-group ext2 in
```

```
Router#sh acc
Extended IP access list ext3
    10 deny icmp 3.3.3.0 0.0.0.255 host 23.1.1.2
    20 deny icmp 3.3.3.0 0.0.0.255 host 12.1.1.2
    30 permit ip any any
```