西安电子科技大学

计算机与网络安全综合实验 课程实验报告

实验名称 配置 ACL

网络与信息安全 学院 2118021 班

姓名 学号

同作者

成 绩

实验日期 2024 年 05 月 22 日

指导教师评语：   指导教师： 年 月 日
实验报告内容基本要求及参考格式
一、实验目的 二、实验所用仪器（或实验环境） 三、实验基本原理及步骤（或方案设计及理论计算） 四、实验数据记录（或仿真及软件设计） 五、实验结果分析及回答问题（或测试环境及测试结果）

ACL 基础

一、实验目的

1. 配置 H3C 路由器基本 ACL。

二、实验要求

1. 熟练掌握网络配置能力。
2. 熟练掌握 ACL 基本配置。

三、实验步骤

由于路由器已配置状态可能会对试验造成干扰，因此为消除这种干扰，在实验前对于两台使用的路由器均首先使用 reset save 清除原有配置，再使用 reboot 命令保存状态并进行重启。

```
<H3C>reset save
The saved configuration file will be erased. Are you sure? [Y/N]:y
Configuration file does not exist!
<H3C>reboot
Start to check configuration with next startup configuration file, please wait.........DONE!
Current configuration may be lost after the reboot, save current configuration? [Y/N]:n
This command will reboot the device. Continue? [Y/N]:y
Now rebooting, please wait...
%Nov 30 04:03:23:670 2022 H3C DEV/5/SYSTEM_REBOOT: System is rebooting now.

System is starting...
```

待路由器重启后，分别将其命名为 H3C-R1 与 H3C-R2。命名时首先使用 system-view 进入系统视图，此后使用 sysname H3C-R1 与 sysname H3C-R2 分别命名两台路由器。

```
<H3C>system-view
System View: return to User View with Ctrl+Z.
[H3C]sysname H3C-R1
[H3C-R1]
```

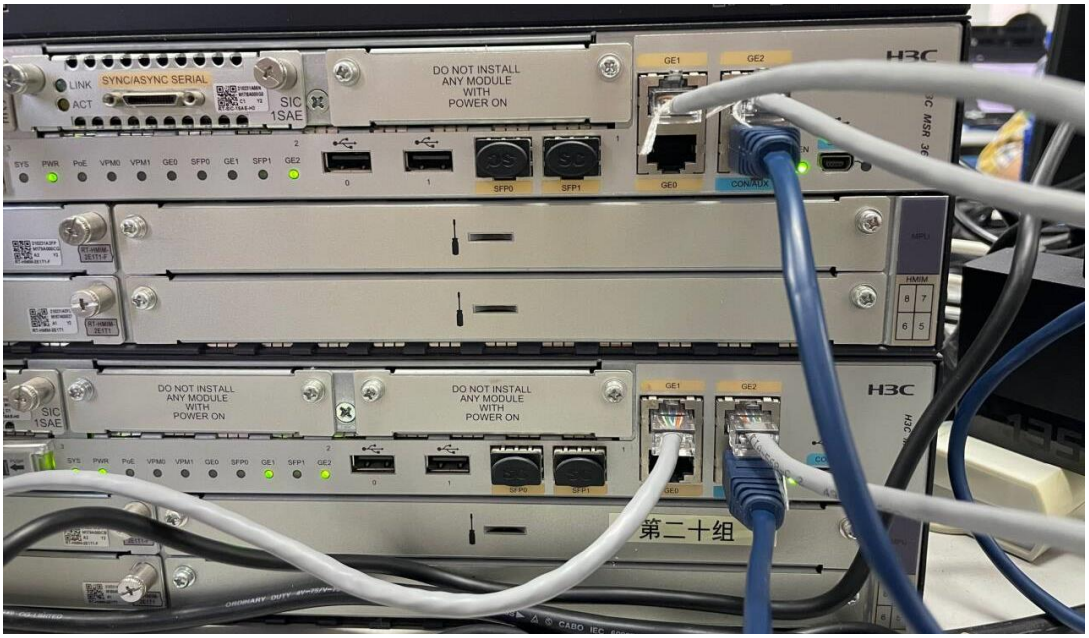
```
<H3C>system-view
System View: return to User View with Ctrl+Z.
[H3C]sysname H3C-R2
[H3C-R2]
```

此后对两台路由器使用到的端口配置 IP 地址。首先对 H3C-R1 的 GE0/1 端口进行配置，使用 interface GigabitEthernet0/1 进入 GE0/1 端口视图，此后使用 ip address 192.168.1.1 24 配置其 IP 地址为 192.168.1.1/24。此后对于 H3C-R1 的 GE0/2 端口与 H3C-R2 的 GE0/1、GE0/2 端口可按照相同的命令进行配置。

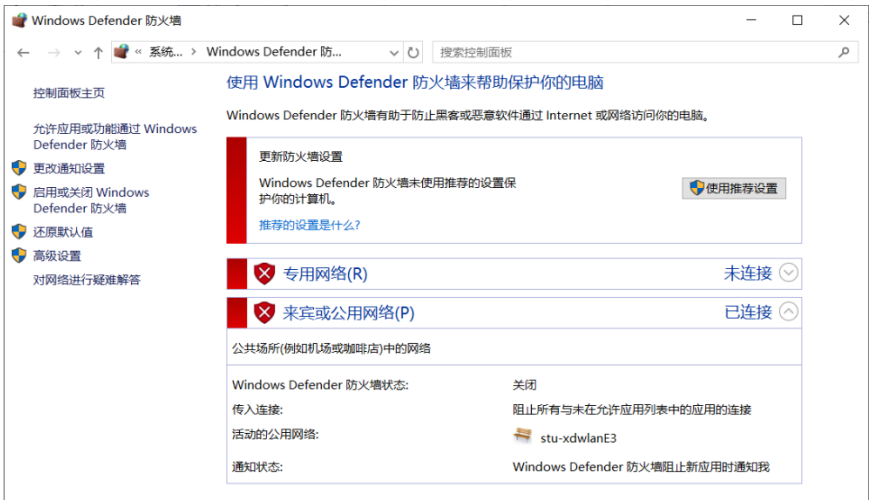
```
[H3C-R1]interface GigabitEthernet0/1
[H3C-R1-GigabitEthernet0/1]ip address 192.168.1.1 24
[H3C-R1-GigabitEthernet0/1]quit
[H3C-R1]interface GigabitEthernet0/2
[H3C-R1-GigabitEthernet0/2]ip address 192.168.2.1 24
[H3C-R1-GigabitEthernet0/2]
```

```
[H3C-R2]interface GigabitEthernet0/2
[H3C-R2-GigabitEthernet0/2]ip address 192.168.2.2 24
[H3C-R2-GigabitEthernet0/2]quit
[H3C-R2]interface GigabitEthernet0/1
[H3C-R2-GigabitEthernet0/1]ip address 192.168.3.1 24
[H3C-R2-GigabitEthernet0/1]quit
```

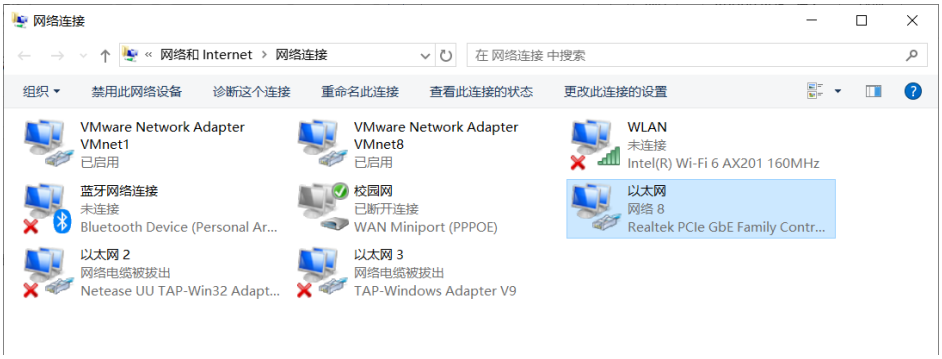
配置后 PC1、PC2 及两台路由器的连接如下：



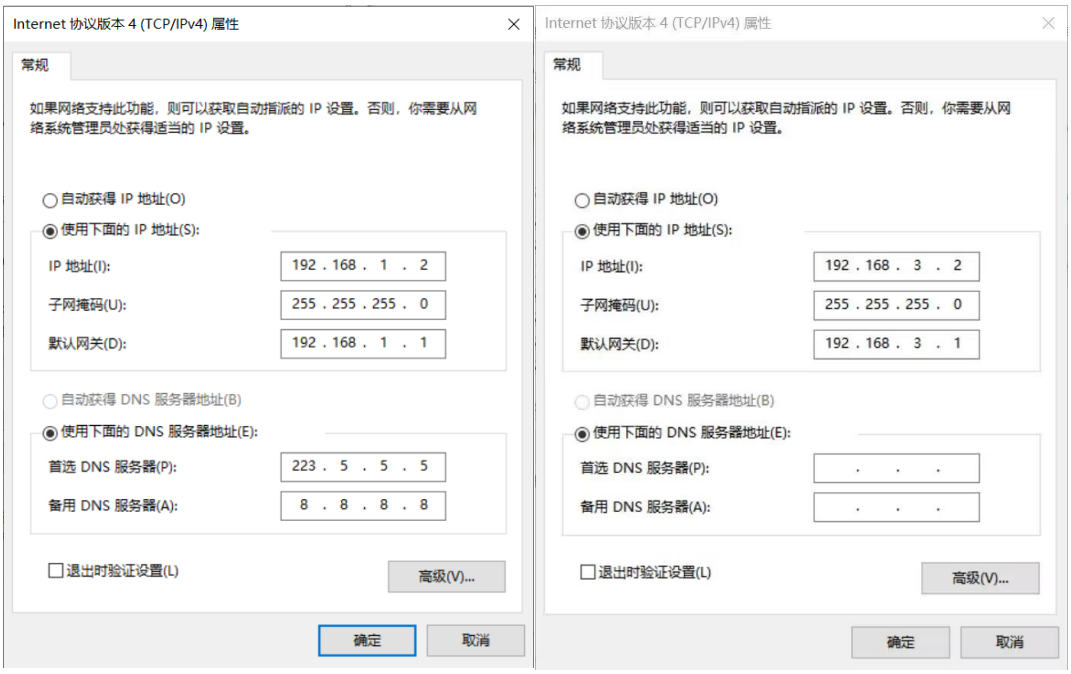
由于防火墙会对实验产生干扰，因此在控制面板中关闭 PC1 与 PC2 的防火墙。



此后在控制面板的网络连接中配置两台 PC 的 IP 地址与默认网关。在网络连接界面，选择以太网属性，配置 IPv4 协议 IP 地址。



由于两相连端口的网络地址必须相同，因此配置 PC1 的 IP 地址为 192.168.1.2/24，其默认网关为与其相连的 H3C-R1 的 GE0/1 端口，即 192.168.1.1。同理配置 PC2 的 IP 地址为 192.168.3.2/24，其默认网关为与其相连的 H3C-R2 的 GE0/1 端口，即 192.168.3.1。配置结果如下（左 PC1，右 PC2）：



此后，在 PC1 上配置通往网络号为 192.168.3.0/24 的默认路由为 192.168.2.2/24，在 PC2 上配置通往网络号为 192.168.1.0/24 的默认路由为 192.168.2.1/24

```
[H3C-R1]ip route-static 192.168.3.0 255.255.255.0 192.168.2.2
```

```
[H3C-R2]ip route-static 192.168.1.0 255.255.255.0 192.168.2.1
```

配置后使用 ip routing 查看其路由表，均可看到配置的默认路由：

```
[H3C-R1]display ip routing

Destinations : 17       Routes : 17

Destination/Mask    Proto   Pre Cost        NextHop         Interface
0.0.0.0/32          Direct  0   0           127.0.0.1       InLoop0
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
192.168.1.0/24      Direct  0   0           192.168.1.1     GE0/1
192.168.1.0/32      Direct  0   0           192.168.1.1     GE0/1
192.168.1.1/32      Direct  0   0           127.0.0.1       InLoop0
192.168.1.255/32    Direct  0   0           192.168.1.1     GE0/1
192.168.2.0/24      Direct  0   0           192.168.2.1     GE0/2
192.168.2.0/32      Direct  0   0           192.168.2.1     GE0/2
192.168.2.1/32      Direct  0   0           127.0.0.1       InLoop0
192.168.2.255/32    Direct  0   0           192.168.2.1     GE0/2
192.168.3.0/24      Static  60  0           192.168.2.2     GE0/2
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
[H3C-R1]
```

```
[H3C-R2]display ip routing

Destinations : 17       Routes : 17

Destination/Mask    Proto   Pre Cost        NextHop         Interface
0.0.0.0/32          Direct  0   0           127.0.0.1       InLoop0
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
192.168.1.0/24      Static  60  0           192.168.2.1     GE0/2
192.168.2.0/24      Direct  0   0           192.168.2.2     GE0/2
192.168.2.0/32      Direct  0   0           192.168.2.2     GE0/2
192.168.2.2/32      Direct  0   0           127.0.0.1       InLoop0
192.168.2.255/32    Direct  0   0           192.168.2.2     GE0/2
192.168.3.0/24      Direct  0   0           192.168.3.1     GE0/1
192.168.3.0/32      Direct  0   0           192.168.3.1     GE0/1
192.168.3.1/32      Direct  0   0           127.0.0.1       InLoop0
192.168.3.255/32    Direct  0   0           192.168.3.1     GE0/1
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
```

此时分别在 PC1 与 PC2 的控制台上使用 ping 命令测试二者之间的连通性，上图为 PC1，下图为 PC2.

```
C:\Users\h'z'f>ping 192.168.3.2

正在 Ping 192.168.3.2 具有 32 字节的数据:
来自 192.168.3.2 的回复: 字节=32 时间<1ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间<1ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间<1ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间<1ms TTL=126

192.168.3.2 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms，最长 = 0ms，平均 = 0ms
```

```
C:\Users\Ghost_N95>ping 192.168.1.2

正在 Ping 192.168.1.2 具有 32 字节的数据:
来自 192.168.1.2 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.2 的回复: 字节=32 时间=1ms TTL=62
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=62
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=62

192.168.1.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 2ms, 平均 = 1ms
```

由结果可知，此时二者之间相互连通。

此后在 H3C-R1 上配置 ACL 对来自 IP 地址为 192.168.1.2 的数据包进行过滤。首先通过 acl basic name PC1 创建名称为 PC1 的 ACL 并进入 ACL 视图，使用 rule deny source 192.168.1.2 0 配置 ACL 对来自 IP 地址为 192.168.1.2 的数据包进行过滤,此后使用 quit 退出 ACL 视图,进入 GE0/1 视图后,使用 packet-filter name PC1 inbound 将 PC1 这一 ACL 应用至 H3C-R1 的 GE0/1 端口。

```
[H3C-R1]acl basic name PC1
[H3C-R1-acl-ipv4-basic-PC1]rule deny source  192.168.1.2 0
[H3C-R1-acl-ipv4-basic-PC1]quit
[H3C-R1]interface GigabitEhternet0/1
                  ^
 % Wrong parameter found at '^' position.
[H3C-R1]interface GigabitEthernet0/1
[H3C-R1-GigabitEthernet0/1]packet-filter name PC1 inbound
[H3C-R1-GigabitEthernet0/1]quit
[H3C-R1]
```

此时可以看到，当应用 ACL 后，原本连通的连接断开。

```
C:\Users\h'z'f>ping 192.168.3.2 -t

正在 Ping 192.168.3.2 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。

192.168.3.2 的 Ping 统计信息:
    数据包: 已发送 = 11, 已接收 = 0, 丢失 = 11 (100% 丢失),
Control-C
^C
```

四、实验结果及分析

1. 实验过程中遇到什么问题，如何解决的？通过该实验有何收获？

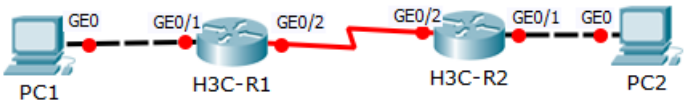
实验时，遇到 PC2 可以 ping 通 PC1，而 PC1 不能 ping 通 PC2 的情况，在排查后发现 PC2 存在第三方防护软件对流量进行了过滤，使得 PC1 不能与 PC2 连通，而 PC2 可以与 PC1 连通。

通过本次实验的操作，我进一步熟悉了对 H3C 交换机的操作命令，并通过实验前后现象的对比，对 ACL 这一对数据包进行过滤的功能有了清晰的认识，熟悉了 ACL 技术的应用方法，进一步理解了通过 ACL 技术实现安全控制的原理与场景。

2. 请按照 PPT 中第 10 页表格的格式，给出你实际实验时所使用的数据。

设备名称	接口名称	IP 地址
H3C-R1	GE0/1	192.168.1.1/24
H3C-R1	GE0/2	192.168.2.1/24
H3C-R2	GE0/1	192.168.3.1/24
H3C-R2	GE0/2	192.168.2.2/24
PC1	GE0	192.168.1.2/24
PC2	GE0	192.168.3.2/24

3. 请使用 Packet Tracer 绘制本实验拓扑图，并且说明，如果 PC1 的 IP 地址改为 192.168.5.2，H3C-R1 的哪个端口要做相应修改，应该改成什么？这时候 PC1 的默认网关要改成什么？



由于相连端口的网络地址应相同，且 PC1 与 H3C-R1 的 E0/1 端口相连，因此 PC1 的 IP 地址改变后，H3C-R1 的 E0/1 端口的 IP 地址需要做出相应修改，应改为 192.168.5.1，PC1 的默认网关应改为 192.168.5.1。

4. 请使用自己的语言对以下实验核心代码做以简单解释

```
[H3C-R1]acl basic name PC1
[H3C-R1-acl-ipv4-basic-PC1]rule deny source 192.168.1.2 0
[H3C-R1-acl-ipv4-basic-PC1]quit
[H3C-R1]interface GigabitEthernet0/1
[H3C-R1-GigabitEthernet0/1]packet-filter name PC1 inbound
[H3C-R1-GigabitEthernet0/1]quit
```

第 1 行 acl basic name PC1 表示创建名为 PC1 的基本 ACL，之后进入 ACL 视图。

第 2 行 rule deny source 192.168.1.2 0 表示配置 PC1 的 ACL 规则：拒绝 IP 地址为 192.168.1.2 的数据包通过。

第 3 行 quit 退出 ACL 视图。

第 4 行 interface GigabitEthernet0/1 表示进入 GE0/1 端口视图。

第 5 行 packe-filter name PC1 inbound 表示将 PC1 这一 ACL 应用在 GE0/1 端口的入方向，从而实现对数据包的过滤。

第 6 行 quit 表示退出 GE0/1 端口视图。

配置高级ACL实现包过滤

一、实验目的

1. 配置 H3C 路由器高级 ACL 及实现包过滤防火墙配置；

2. 熟悉 ACL 查看、监测和调试的相关命令；

二、实验要求

1. 2 台具有 4 个以太网接口的路由器；

2. 3 台装有 Windows 系列操作系统的 PC（台式机或笔记本）；

3. 4 条双绞跳线（交叉线）；

三、实验步骤

1. 配置高级 ACL 实现包过滤

由于两个实验连续进行，因此需要首先清除上一步的配置，此处和实验一一样首先将 R1 的配置清空并重启，R2 不需要，因为上一步并没有针对 R2 设置包拦截。

```
<H3C-R1>reset save
The saved configuration file will be erased. Are you sure? [Y/N]:y
Configuration file does not exist!
<H3C-R1>reboot
Start to check configuration with next startup configuration file, please wait.........DONE!
Current configuration may be lost after the reboot, save current configuration? [Y/N]:n
This command will reboot the device. Continue? [Y/N]:y
Now rebooting, please wait...
%Nov 30 04:58:40:418 2022 H3C-R1 DEV/5/SYSTEM_REBOOT: System is rebooting now.
```

之后重新配置 R1 的 ip 地址

```
<H3C>system-view
System View: return to User View with Ctrl+Z.
[H3C]sysname H3C-R1
[H3C-R1]interface GigabitEthernet0/1
[H3C-R1-GigabitEthernet0/1]ip address 192.168.1.1 24
[H3C-R1-GigabitEthernet0/1]quit
[H3C-R1]interface GigabitEthernet0/2
[H3C-R1-GigabitEthernet0/2]ip address 192.168.2.1 24
[H3C-R1-GigabitEthernet0/2]
```

然后将 R2 新增连接设备 PC3 的 E0/0 端口进行配置，并设置 PC3 的 ip 地址和网关。

```
[H3C-R2]interface GigabitEthernet0/0
[H3C-R2-GigabitEthernet0/0]ip address 192.168.4.1 24
```

之后还需要设置静态路由，因为实验需要在设置高级 ACL 后验证 PC1 不可 ping 通 PC2 但可 ping 通 PC3，如果不设置静态路由，则 PC1 也不能 ping 通 PC3，无法达到实验目的。

```
[H3C-R1]ip route-static 192.168.3.0 255.255.255.0 192.168.2.2
[H3C-R1]ip route-static 192.168.4.0 255.255.255.0 192.168.2.2
[H3C-R1]
```

在此时首先测试一下 PC1 与 PC2 和 PC3 的联通性，可以看到 PC1 与 PC2 和 PC3 均联通。

```
C:\Users\h'z'f>ping 192.168.3.2

正在 Ping 192.168.3.2 具有 32 字节的数据:
来自 192.168.3.2 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.3.2 的回复: 字节=32 时间=2ms TTL=126

192.168.3.2 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms，最长 = 2ms，平均 = 1ms

C:\Users\h'z'f>ping 192.168.4.2

正在 Ping 192.168.4.2 具有 32 字节的数据:
来自 192.168.4.2 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=1ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=1ms TTL=126

192.168.4.2 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms，最长 = 1ms，平均 = 1ms

C:\Users\h'z'f>
```

之后添加 ACL 规则。首先创建 pc1-pc2，拒绝从源 IP 192.168.1.2 到目的地址范围 192.168.3.0——192.168.3.255 范围内的 IP 流量。这对应着 PC1 无法发送 IMCP 报文到 PC2，即无法 ping 通。

之后创建 pc2telnet 的 ACL。首先添加允许源 IP 地址 192.168.1.2 到目的端口 23（telnet 端口）的 TCP 流量，并追加拒绝任何源地址到目的端口 23 的 TCP 流量规则。这对应着只有 PC1 可以与 R1 建立链接。

```
[H3C-R1]acl advanced name pc1-pc2
[H3C-R1-acl-ipv4-adv-pc1-pc2]rule deny ip source 192.168.1.2 0 destination 192.1
68.3.0 0.0.0.255
[H3C-R1-acl-ipv4-adv-pc1-pc2]quit
[H3C-R1]acl advanced name pc2telnet
[H3C-R1-acl-ipv4-adv-pc2telnet]rule permit tcp source 192.168.1.2 0 destination-
port eq 23
[H3C-R1-acl-ipv4-adv-pc2telnet]rule deny tcp source any destination-port eq 23
[H3C-R1-acl-ipv4-adv-pc2telnet]display this
#
acl advanced name pc2telnet
 rule 0 permit tcp source 192.168.1.2 0 destination-port eq telnet
 rule 5 deny tcp destination-port eq telnet
#
return
[H3C-R1-acl-ipv4-adv-pc2telnet]
```

之后将 ACL pc1-pc2 应用到 E0/2 端口的出站流量，将 ACL pc2telnet 应用到 E0/1 和 E0/2 的入站流量。

```
[H3C-R1-acl-ipv4-adv-pc2telnet]quit
[H3C-R1]interface GigabitEthernet0/2
[H3C-R1-GigabitEthernet0/2]packet-filter name pc1-pc2 outbound
[H3C-R1-GigabitEthernet0/2]packet-filter name pc2telnet outbound
[H3C-R1-GigabitEthernet0/2]packet-filter name pc2telnet inbound
[H3C-R1-GigabitEthernet0/2]undo packet-filter name pc2telnet outbound
[H3C-R1-GigabitEthernet0/2]packet-filter name pc2telnet inbound
[H3C-R1-GigabitEthernet0/2]quit
[H3C-R1]interface GigabitEthernet 0/1
[H3C-R1-GigabitEthernet0/1]packet-filter name pc2telnet inbound
[H3C-R1-GigabitEthernet0/1]quit
[H3C-R1]
```

2. 从 PC1 上，使用 ping 命令访问 PC2 的 IP 地址 192.168.3.2，测试连通性

可以看到，由于 ACL pc1-pc2 的使用，原本可以 ping 通的 PC1 和 PC2 之间现在无法 ping 通，ICMP 报文被防火墙阻拦。

```
C:\Users\h'z'f>ping 192.168.3.2

正在 Ping 192.168.3.2 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.3.2 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 0，丢失 = 4 (100% 丢失)，
```

3. 从 PC1 上，使用 ping 命令访问 PC3 的 IP 地址 192.168.4.2，测试连通性

可以看到，PC1 和 PC3 之间由于没有 ACL 的阻碍，且正确配置了端口 IP 和路由，它们仍然可以 ping 通。

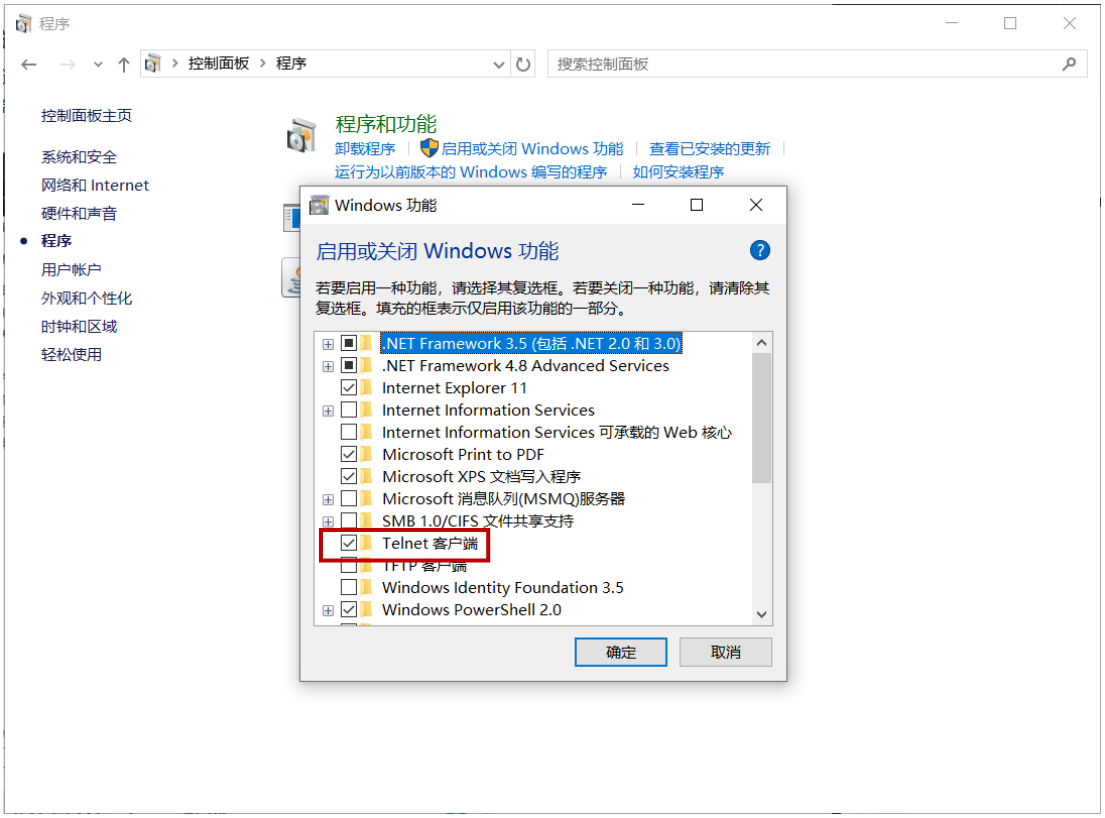
```
C:\Users\h'z'f>ping 192.168.4.2

正在 Ping 192.168.4.2 具有 32 字节的数据:
来自 192.168.4.2 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=2ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=3ms TTL=126
来自 192.168.4.2 的回复: 字节=32 时间=2ms TTL=126

192.168.4.2 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 2ms，最长 = 3ms，平均 = 2ms
```

4. 在 H3C-R2 上测试使用 telnet 访问 H3C-R1

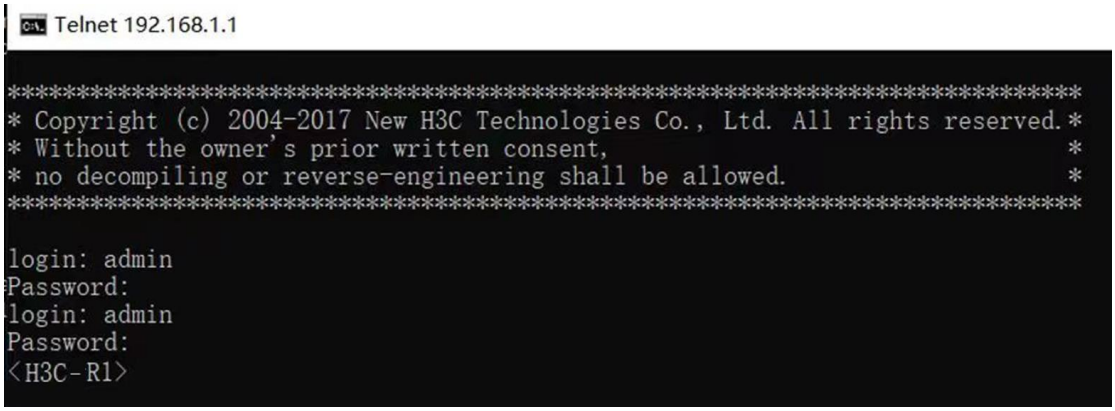
最后测试 R2 使用 telnet 访问 R1，此处由于 ACL pc2telnet 在 E0/1 和 E0/2 端口控制入站流量，且 R2 与 R1 相连的 E0/2 端口的 ip 为 192.168.2.2，并非可以通过的 PC1 ip 192.168.1.2，因此 telnet 被拦截。

```
<H3C-R2>telnet 192.168.2.1
Trying 192.168.2.1 ...
Press CTRL+K to abort
Connected to 192.168.2.1 ...
Failed to connect to the remote host!
```

之后测试 PC1 是否可以使用 telnet 访问 R1。在 PC1 中，首先需要安装 telnet 程序。



之后在 cmd 中使用 telnet 192.168.1.1 访问 R1，可以发现连接成功。



五、实验结果及分析

1. 整个实验过程中遇到什么问题（有截图最好），如何解决的？通过该实验有何收获？

在做完上一个实验时，并没有清空 R1 的 ACL 规则，导致保留了上一个实验的 ACL，使得在刚开始测试连通性时，PC1 与 PC2 就不相通。最后通过删除原有配置解决了这一问题。

通过本次实验的操作，我进一步熟悉了对 H3C 交换机的操作命令，并通过实验前后现象的对比，对高级 ACL 实现包过滤的功能有了清晰的认识，更加熟悉了 ACL 技术的应用方法，进一步理解了通过 ACL 技术实现安全控制的原理与场景。