

网络技术与应用课程实验报告

实验名称：防火墙配置

学号： 2211489 姓名： 冯佳明 专业： 物联网工程

一、实验要求：

防火墙实验在虚拟仿真环境下完成，要求如下：

1. 了解包过滤防火墙的基本配置方法、配置命令和配置过程。
2. 利用标准 ACL，将防火墙配置为只允许某个网络中的主机访问另一个网络。
3. 利用扩展 ACL，将防火墙配置为拒绝某个网络中的某台主机访问网络中的 Web 服务器。
4. 将防火墙配置为允许内网用户自由地向外网发起 TCP 连接，同时可以接收外网发回的 TCP 应答数据包。但是，不允许外网的用户主动向内网发起 TCP 连接。

二、实验步骤

（一）了解包过滤防火墙的基本配置方法、配置命令和配置过程

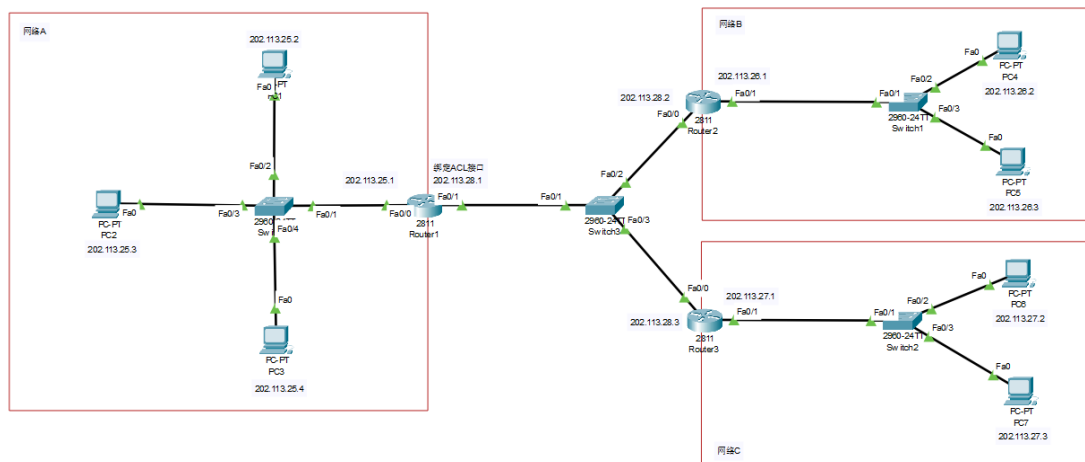
包过滤防火墙是一种基于网络层（第三层）的访问控制机制，它通过检查流经它的数据包的源 IP 地址、目的 IP 地址、协议类型（如 TCP、UDP、ICMP 等）、端口号等信息来决定是否允许该数据包通过。在 Cisco 路由器中，这种功能通常是通过访问控制列表（ACLs）来实现的。

在 Cisco 中，控制访问列表的匹配顺序为“自上而下，依次匹配”，默认为拒绝。所以在配置过滤规则时，ACL 语句的顺序很重要，数据包只有在跟第一个判断条件不匹配时，才能被交给 ACL 中下一个条件语句进行比较。

在本次实验中会分别对标准访问控制列表和扩展访问控制列表进行配置，标准访问控制列表的表号范围为 1-99，扩展访问控制列表号的范围是 100-199。

（二）利用标准 ACL，将防火墙配置为只允许某个网络中的主机访问另一个网络

1. 网络拓扑结构如下图所示



2. 配置各主机 IP 地址，子网掩码，默认网关

	IP 地址	子网掩码	默认网关
PC1	202.113.25.2	255.255.255.0	202.113.25.1
PC2	202.113.25.3	255.255.255.0	202.113.25.1
PC3	202.113.25.4	255.255.255.0	202.113.25.1
PC4	202.113.26.2	255.255.255.0	202.113.26.1
PC5	202.113.26.	255.255.255.0	202.113.26.1
PC6	202.113.27.2	255.255.255.0	202.113.27.1
PC7	202.113.27.3	255.255.255.0	202.113.27.1

3. 配置路由器端口并激活，使用 rip 协议，动态配置三个路由器的路由表项

Router1	IP 地址	子网掩码
Fa0/0	202.113.25.1	255.255.255.0
Fa0/1	202.113.28.1	255.255.255.0

```
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 202.113.25.1 255.255.255.0
Router(config-if)#ip address 202.113.25.1 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 202.113.28.1 255.255.255.0
Router(config-if)#
```

```
Router(config)#router rip
Router(config-router)#network 202.113.25.0
Router(config-router)#network 202.113.28.0
```

Router2	IP 地址	子网掩码
Fa0/0	202.113.28.2	255.255.255.0
Fa0/1	202.113.26.1	255.255.255.0

```
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 202.113.28.2 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 202.113.26.1 255.255.255.0
Router(config-if)#
```

```
Router(config)#router rip
Router(config-router)#network 202.113.28.0
Router(config-router)#network 202.113.26.0
```

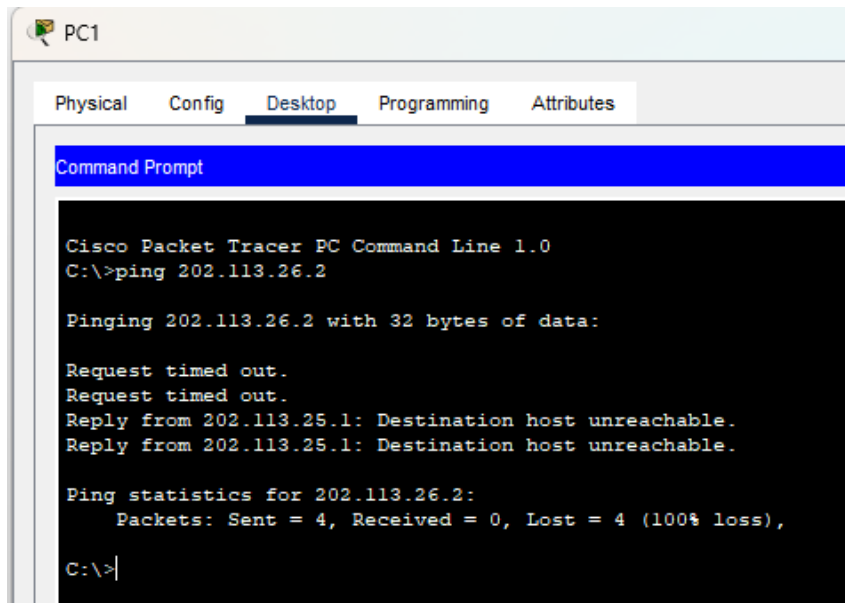
Router3	IP 地址	子网掩码
Fa0/0	202.113.28.3	255.255.255.0
Fa0/1	202.113.27.1	255.255.255.0

```
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 202.113.28.3 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 202.113.27.1 255.255.255.0
Router(config-if)#
```

```
Router(config)#router rip
Router(config-router)#network 202.113.27.0
Router(config-router)#network 202.113.28.0
```

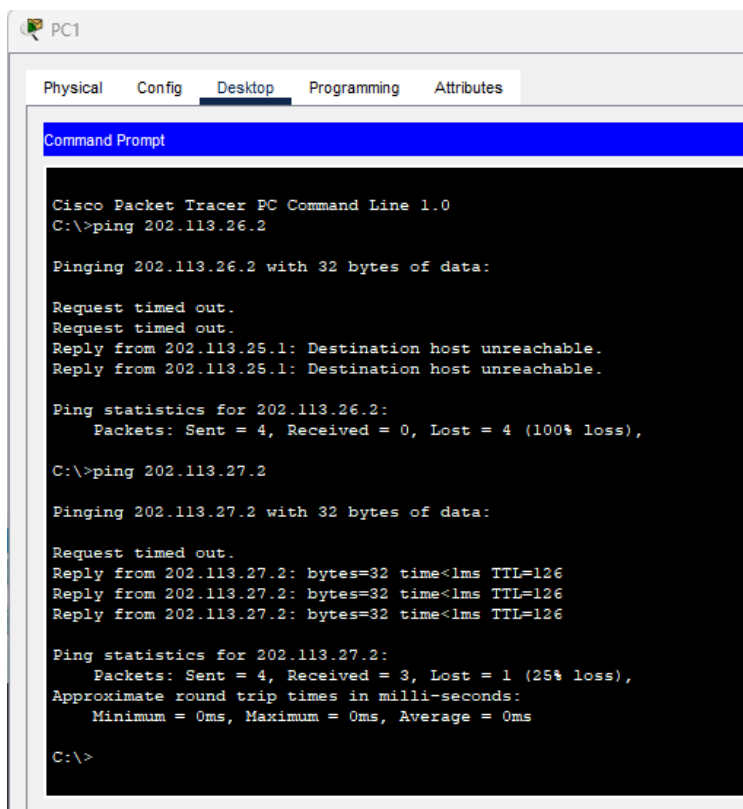
4. 测试首先测试在没有设置对应的 ACL 的情况下的网络的连通性

(1) 网络 A 中主机 PC1 ping 网络 B 中主机 PC4 (202.113.26.2)



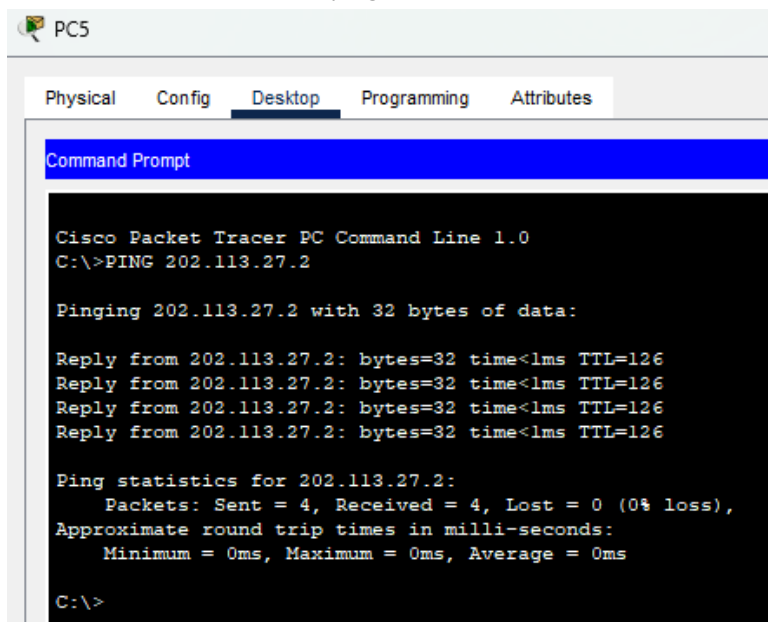
前两个请求丢失原因：第一次获得到 router1 的 MAC 地址，第二次获得到 router2 的 MAC 地址

(2) 网络 A 中主机 PC1 ping 网络 C 中主机 PC6 (202.113.27.2)



第一个请求丢失原因：获得到 router3 的 MAC 地址（router1 的地址在 ping PC4 时已经获得）

(3) 网络 B 中主机 PC5 ping 网络 C 中主机 PC6 (202.113.27.2)



```
PC5
Physical Config Desktop Programming Attributes
Command Prompt

Cisco Packet Tracer PC Command Line 1.0
C:\>PING 202.113.27.2

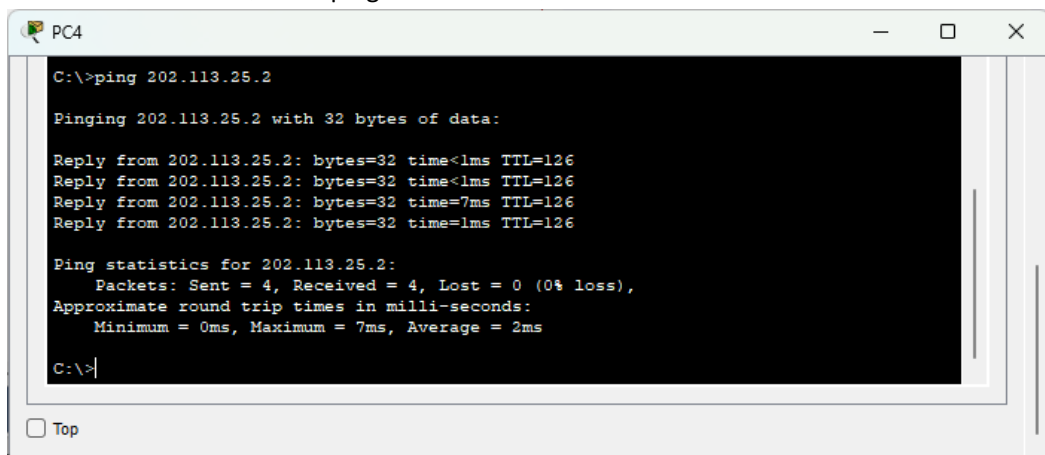
Pinging 202.113.27.2 with 32 bytes of data:

Reply from 202.113.27.2: bytes=32 time<1ms TTL=126
Reply from 202.113.27.2: bytes=32 time<1ms TTL=126
Reply from 202.113.27.2: bytes=32 time<1ms TTL=126
Reply from 202.113.27.2: bytes=32 time<1ms TTL=126

Ping statistics for 202.113.27.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

(4) 网络 B 中主机 PC4 ping 网络 A 中主机 PC1 (202.113.25.2)



```
PC4
C:\>ping 202.113.25.2

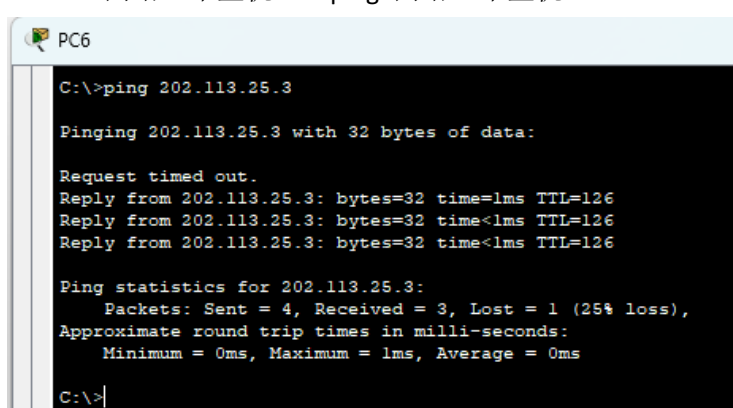
Pinging 202.113.25.2 with 32 bytes of data:

Reply from 202.113.25.2: bytes=32 time<1ms TTL=126
Reply from 202.113.25.2: bytes=32 time<1ms TTL=126
Reply from 202.113.25.2: bytes=32 time=7ms TTL=126
Reply from 202.113.25.2: bytes=32 time=1ms TTL=126

Ping statistics for 202.113.25.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 2ms

C:\>
```

(5) 网络 C 中主机 PC6 ping 网络 A 中主机 PC2 (202.113.25.3)



```
PC6
C:\>ping 202.113.25.3

Pinging 202.113.25.3 with 32 bytes of data:

Request timed out.
Reply from 202.113.25.3: bytes=32 time=1ms TTL=126
Reply from 202.113.25.3: bytes=32 time<1ms TTL=126
Reply from 202.113.25.3: bytes=32 time<1ms TTL=126

Ping statistics for 202.113.25.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

5. 配置标准 ACL，将防火墙配置为只允许网络 B 中的主机访问网络 A

- (1) 实现方法：配置路由器访问列表（ACL），使路由器实现防火墙功能，对进入 fa0/1 的数据报进行检查和过滤
- (2) 使用命令 `access-list 6 permit 202.113.26.0 0.0.0.255` 建立一个标号为 6 的标准 ACL，允许来自 IP 地址范围 202.113.26.0 到 202.113.26.255 的所有数据包通过 ACL
- (3) 随后用 `access-list 6 deny any` 拒绝所有其他网络发来的数据报，`deny any` 表示拒绝任何来源的数据包，因为没有指定源 IP 地址或范围，所以不管数据包的源 IP 是什么，都会被拒绝
- (4) 根据 ACL 规则设置的先后顺序，该 ACL 实现了允许特定 IP 地址范围（202.113.26.0 到 202.113.26.255）的数据包通过，同时拒绝了来自任何其他 IP 地址的数据包
- (5) 进入 fa0/1 接口配置模式，利用 `ip access-group 6 in` 将 6 号 ACL 绑定在 fa0/1 的入站上，实现网络 B 的主机去通过 fa0/1 接口去访问网络 A，而拒绝网络 C 通过 fa0/1 访问网络 A

```
Router1
Router(config)#
Router(config)#
Router(config)#
Router(config)#
Router(config)#access-list 6 permit 202.113.26.0 0.0.0.255
Router(config)#access-list 6 deny any
Router(config)#
Router(config)#interface fa0/1
Router(config-if)#ip access-group 6 in
Router(config-if)#
```

6. 再次测试网络连通性

- (1) 网络 B 中主机 PC4 ping 网络 A 中主机 PC1（202.113.25.2）
网络 B 中的主机仍然可以访问网络 A 中的主机

```
PC4
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 7ms, Average = 2ms

C:\>ping 202.113.25.2

Pinging 202.113.25.2 with 32 bytes of data:

Reply from 202.113.25.2: bytes=32 time=8ms TTL=126
Reply from 202.113.25.2: bytes=32 time<1ms TTL=126
Reply from 202.113.25.2: bytes=32 time<1ms TTL=126
Reply from 202.113.25.2: bytes=32 time<1ms TTL=126

Ping statistics for 202.113.25.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 8ms, Average = 2ms

C:\>
```

(2) 网络 C 中主机 PC6 ping 网络 A 中主机 PC2 (202.113.25.3)

网络 C 中的主机向网络 A 发送 ping 命令时, 显示 Destination host unreachable
原因为: 数据报 2 发送到 router0 的 fa0/1 接口后, router0 会检查数据报的源 ip 地址, 由于源 ip 地址不在之前设置的允许访问的 ip 地址范围内, 数据报被丢弃

```
PC6
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 202.113.25.3

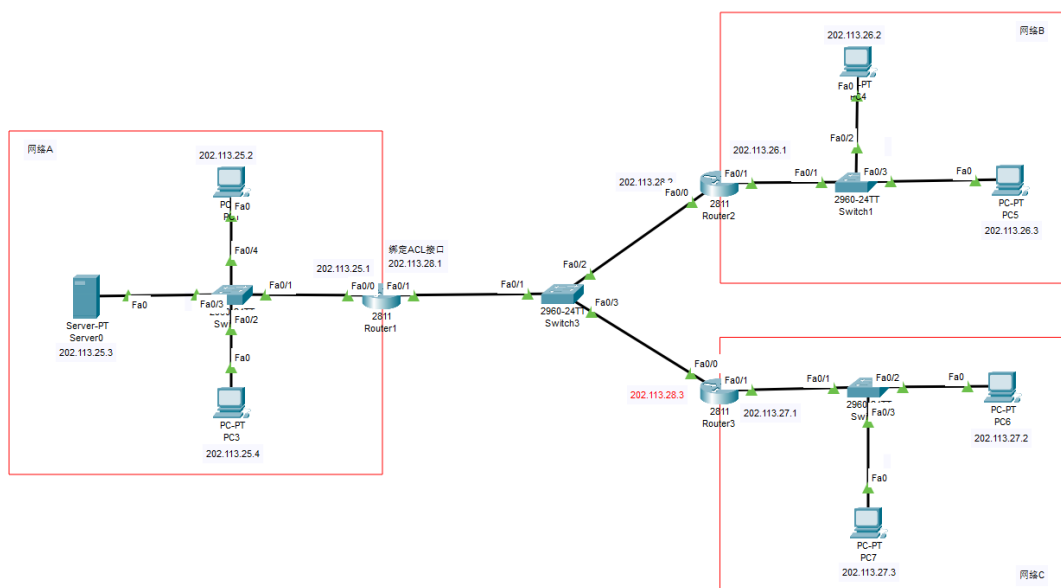
Pinging 202.113.25.3 with 32 bytes of data:

Reply from 202.113.28.1: Destination host unreachable.
Reply from 202.113.28.1: Destination host unreachable.
Reply from 202.113.28.1: Destination host unreachable.
Reply from 202.113.28.1: Destination host unreachable.

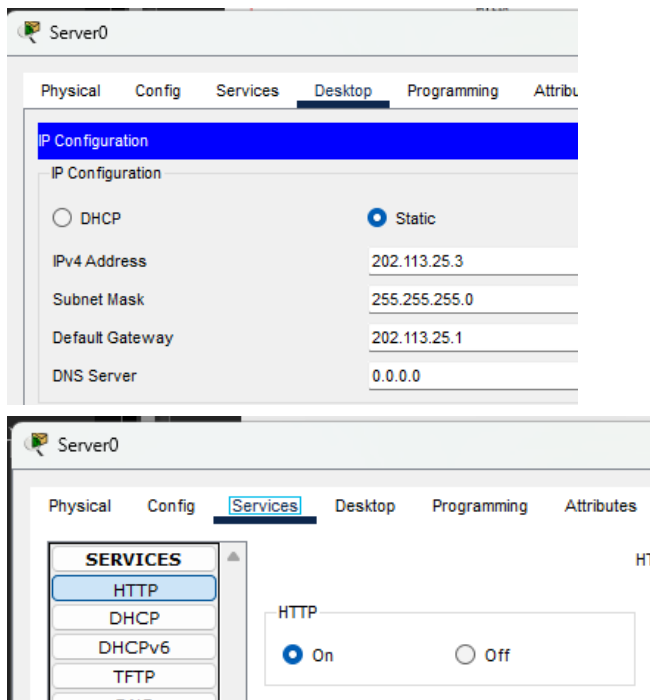
Ping statistics for 202.113.25.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

(三) 利用扩展 ACL, 将防火墙配置为拒绝某个网络中的某台主机访问网络中的 Web 服务器

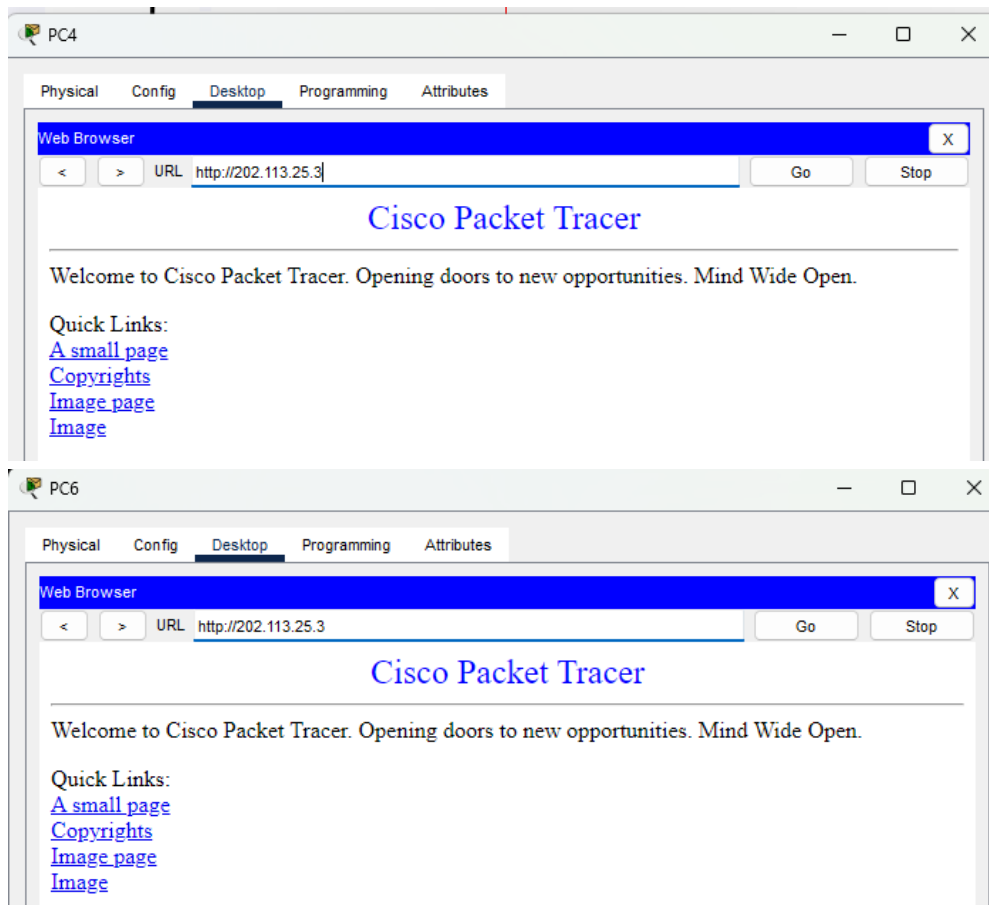
1. 网络拓扑结构如下图所示



2. 配置各主机 IP 地址, 子网掩码, 默认网关: 与之前配置 ACL 步骤一致, 在此不再赘述
3. 配置路由器端口并激活, 使用 rip 协议, 动态配置三个路由器的路由表项: 与之前配置 ACL 步骤一致, 在此不再赘述
4. 配置内网服务器 IP, 并打开服务器的 HTTP 功能



5. 测试没有设置扩展 ACL 时，网络的连通性：分别使用网络 B 中的 PC4 和网络 C 中的 PC6 访问内网服务器，效果如下图所示，可以看到在没有设置扩展 ACL 时，可以访问到内网服务器



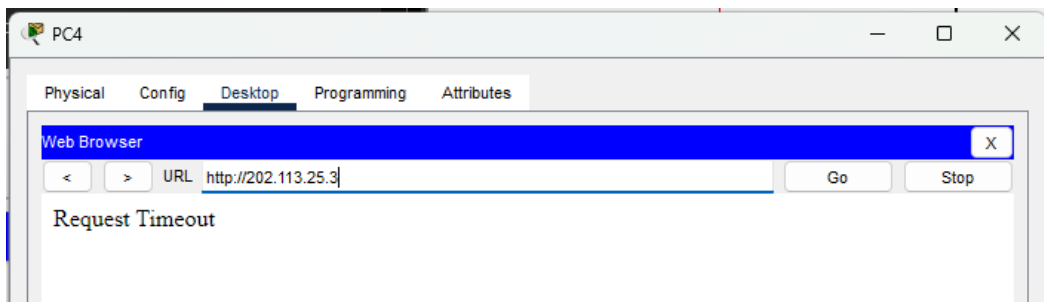
6. 配置扩展 ACL

- (1) 使用 `access-list 106 deny tcp host 202.113.26.2 host 202.113.25.3 eq 80` 命令。建立一个标号为 106 的扩展 ACL，添加规则-抛弃 ip 地址为 202.113.26.2 目的地址为 202.113.25.3 目的端口号为 80 的 TCP 数据包
- (2) 使用 `access-list 106 permit ip any any` 命令，允许任何来源的 IP 地址和任何目的 IP 地址的所有类型的 IP 数据包通过 ACL
- (3) 进入 fa0/1 端口配置模式，利用 `ip access-group 106 in` 将 106 好 ACL 绑定到 fa0/1 的入站上

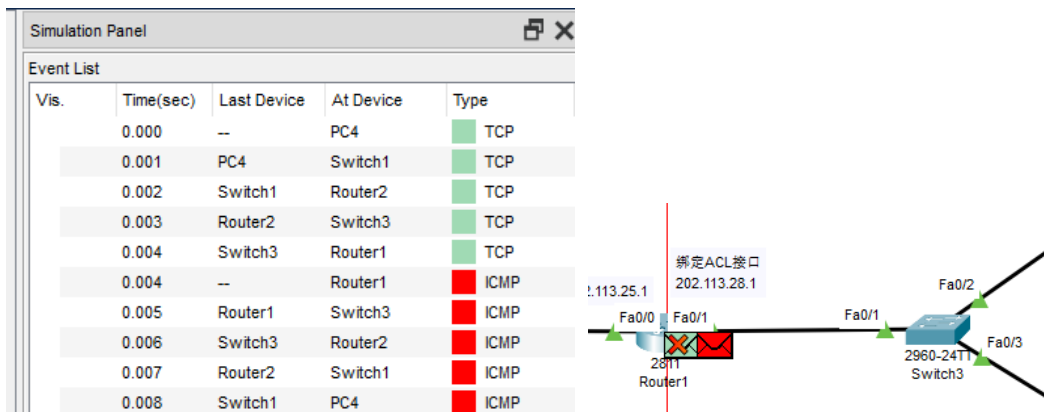
```
Router(config)#access-list 106 deny tcp host 202.113.26.2 host 202.113.25.3 eq 80
Router(config)#
Router(config)#access-list 106 permit ip any any
Router(config)#
Router(config)#interface fa0/1
Router(config-if)#ip access-group 106 in
```

7. 对配置了 ACL 后的网络进行连通性测试

- (1) 使用网络 B 中的 PC4 访问内网服务器，效果如下图所示，可以看到，出现了请求超时的现象，即主机 PC4 对内网服务器的 http 请求被路由器的 fa0/1 端口阻拦



使用模拟方式观察数据包传递过程，可以明显看到数据包在到达路由器后被阻拦，同时返回 ICMP 报文给主机 PC4



- (2) 考虑到之前添加的扩展 ACL 规则为，抛弃源 IP 地址为 202.113.26.2、目的地址为 202.113.25.3 的 TCP 数据包，那理论上来说，使用 PC4 ping 内网服务器，应该是可以 ping 通的，因为 ping 命令使用的是 ICMP 报文，不会被拦截。运行结果如下图所示，证明分析正确


```
PC4

C:\>ping 202.113.25.3

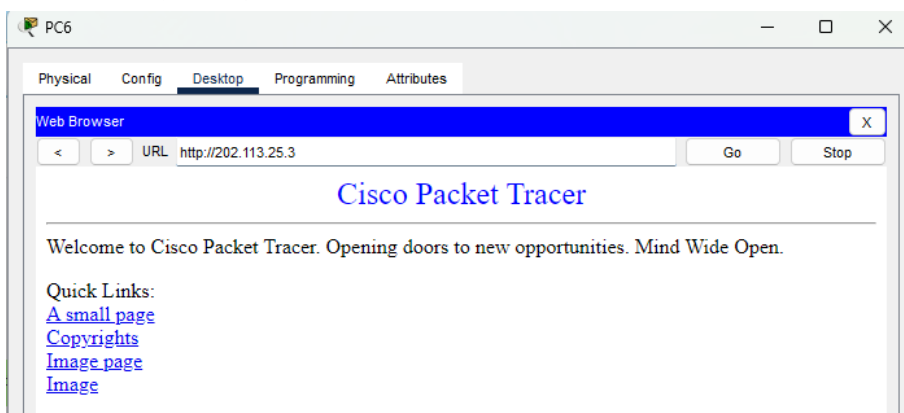
Pinging 202.113.25.3 with 32 bytes of data:

Reply from 202.113.25.3: bytes=32 time=10ms TTL=126
Reply from 202.113.25.3: bytes=32 time<1ms TTL=126
Reply from 202.113.25.3: bytes=32 time<1ms TTL=126
Reply from 202.113.25.3: bytes=32 time<1ms TTL=126

Ping statistics for 202.113.25.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms

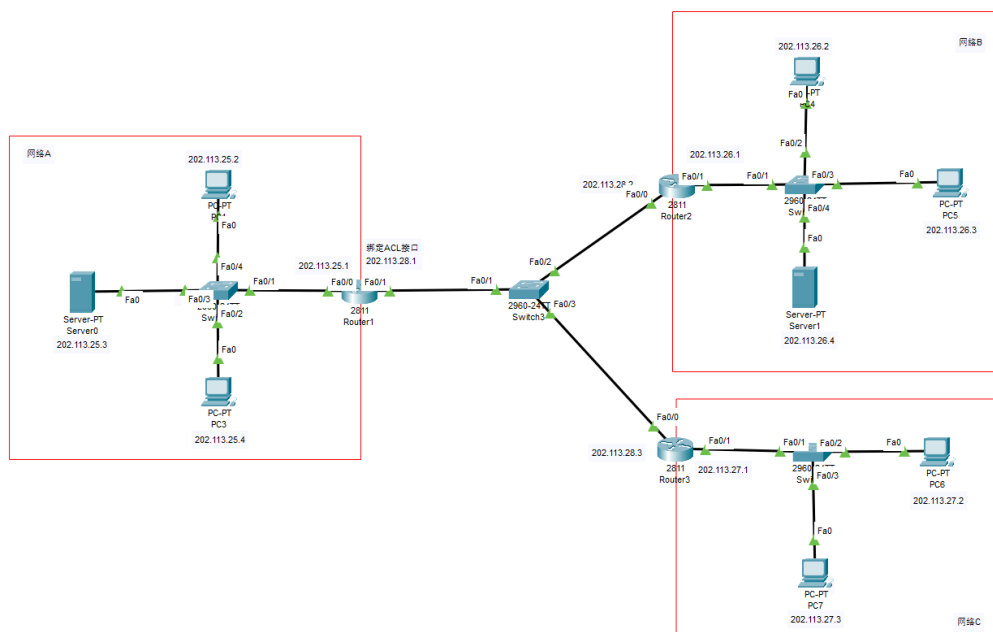
C:\>
```

(3) 使用网络 C 中的 PC6 访问内网服务器，效果如下图所示，可以看到在能够成功访问内网服务器

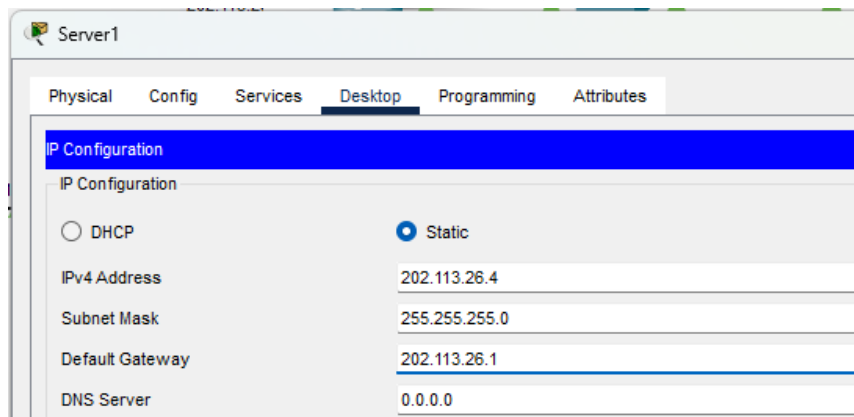


(四) 将防火墙配置为允许内网用户自由地向外网发起 TCP 连接，同时可以接收外网发回的 TCP 应答数据包。但是，不允许外网的用户主动向内网发起 TCP 连接。

1. 实验拓扑结构如下图所示，在网络 B 中增加一台服务器 server1



2. 配置外网服务器 IP 地址并启动 HTTP 服务



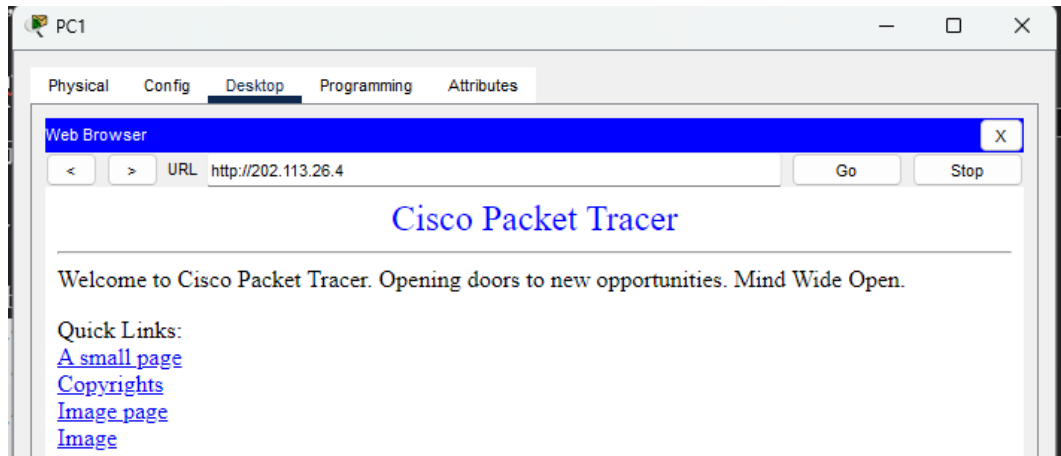
3. 配置扩展 ACL

- (1) 使用 `access-list 108 permit tcp any 202.113.25.0 0.0.0.255 established` 命令，允许任何源 IP 地址，目的 IP 地址为 202.113.25.0 到 202.113.25.255 范围内的 TCP 数据包，并且 TCP 连接状态为 `established`，表示可以接收 TCP 应答数据包
- (2) 使用 `access-list 108 deny tcp any 202.113.25.0 0.0.0.255` 命令，拒绝任何源 IP 地址，目的 IP 地址为 202.113.25.0 到 202.113.25.255 范围内的 TCP 数据包
- (3) 使用 `access-list 108 permit ip any any` 命令，允许任何源 IP、目的 IP 地址的所有 IP 数据包通过 ACL
- (4) 进入 `fa0/1` 端口配置模式，利用 `ip access-group 108 in` 将 108 号 ACL 绑定到 `fa0/1` 的入站上
- (5) 上述的 ACL 规则允许内部网络的主机访问外部网络上的 Web 服务器，同时可以接收外网发回的 TCP 应答数据包。但不允许外网的用户主动向内网发起 TCP 连接

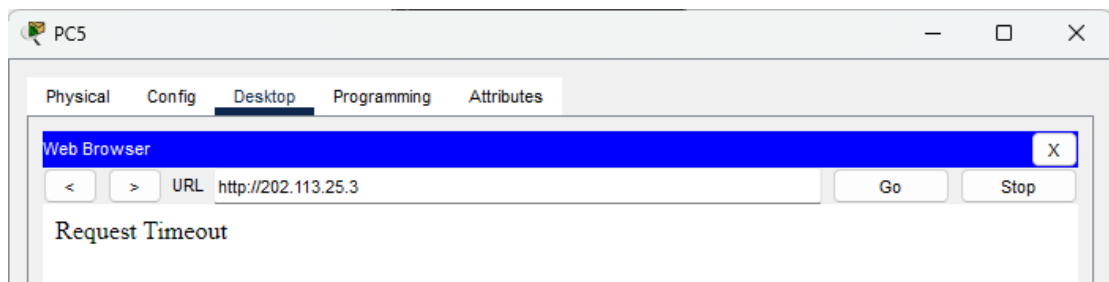
```
Router(config)#access-list 108 permit tcp any 202.113.25.0 0.0.0.255 established
Router(config)#
Router(config)#access-list 108 deny tcp any 202.113.25.0 0.0.0.255
Router(config)#
Router(config)#access-list 108 permit ip any any
Router(config)#
Router(config)#interface fa0/1
Router(config-if)#ip access-group 108 in
Router(config-if)#
```

4. 对配置了 ACL 后的网络进行连通性测试

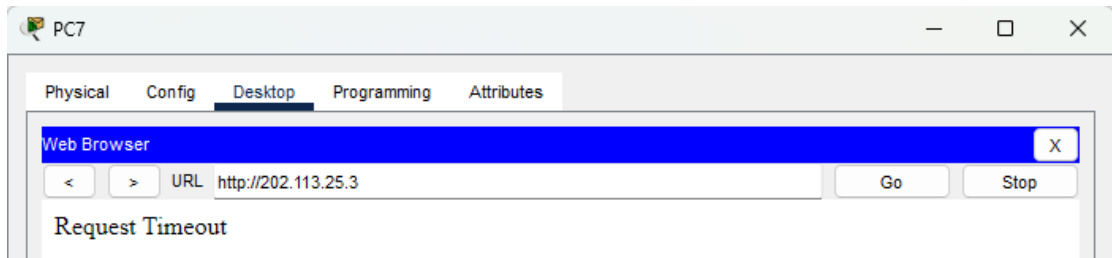
- (1) 使用内网主机 PC1 访问外网服务器



(2) 使用外网 B 中主机 PC5 访问内网服务器



(3) 使用外网 C 中主机 PC7 访问内网服务器



(4) 通过上述的访问服务器的测试，可以验证本次实验的 ACL 配置正确

三、实验总结

通过这次实验，我深刻理解到了 ACL 作为网络安全策略工具的重要性，以及它在实际网络部署中的灵活性。学会了如何根据不同场景需求设计合理的访问控制策略，深刻认识到在设置 ACL 规则时需要特别注意规则的顺序，因为 ACL 是按照从上到下的顺序匹配的，一旦某个规则匹配成功，则不再检查后续规则。