

配置常用的IP ACL

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[配置](#)

[允许选定主机访问网络](#)

[拒绝选定主机访问网络](#)

[允许访问一个范围内的连续 IP 地址](#)

[拒绝 Telnet流量 \(TCP、端口 23 \)](#)

[只允许内部网络发起 TCP 会话](#)

[拒绝 FTP 数据流 \(TCP、端口 21 \)](#)

[允许 FTP 数据流 \(主动 FTP \)](#)

[允许 FTP 数据流 \(被动 FTP \)](#)

[允许 ping \(ICMP\)](#)

[允许 HTTP、Telnet、Mail、POP3、FTP](#)

[允许 DNS](#)

[允许路由更新](#)

[根据 ACL 调试数据流](#)

[Mac 地址过滤](#)

[验证](#)

[故障排除](#)

[相关信息](#)

简介

本文档提供了常用 IP 访问控制列表 (ACL) 的配置示例，ACL 将根据以下内容过滤 IP 数据包：

- 源地址
- 目的地址
- 数据包类型
- 以上各项的任意组合

为了过滤网络数据流，ACL 会控制是转发路由数据包还是将其阻止在路由器接口外。您的路由器是否检查每数据包为了确定转发或丢弃根据您在ACL内指定的标准的数据包。ACL 标准包括：

- 数据流的源地址
- 数据流的目标地址
- 上层协议

完成这些步骤为了修建ACL，在本文的示例显示：

1. 创建 ACL。
2. 将 ACL 应用到某个接口上。

IP ACL是permit的连续的收藏并且拒绝适用于IP数据包的情况。路由器将根据 ACL 中的条件逐个测试数据包。

第一个匹配的条件将决定 Cisco IOS® 软件是接受还是拒绝数据包。因为 Cisco IOS 软件将在匹配第一个条件之后停止测试，所以条件的顺序至关重要。如果没有任何条件匹配，路由器将根据一个隐式 deny all 子句拒绝数据包。

以下是可在 Cisco IOS 软件中配置的 IP ACL 示例：

- 标准 ACL
- 扩展 ACL
- 动态 (锁和密钥) ACL
- IP 命名 ACL
- 自反 ACL
- 使用时间范围的基于时间的 ACL
- 带有注释的 IP ACL 条目
- 基于上下文的 ACL
- 身份验证代理
- Turbo ACL
- 基于时间的分布式 ACL

本文档讨论一些常用的标准 ACL 和扩展 ACL。有关 Cisco IOS 软件支持各类 ACL 以及如何配置和编辑 ACL 的详细信息，请参阅[配置 IP 访问列表](#)。

标准 ACL 的命令语法格式为 **access-list access-list-number {permit|deny} {host|source source-wildcard|其中任一}**。

标准ACL比较IP信息包的源地址对在ACL为了控制流量配置的地址。

延长的ACL比较IP信息包的源地址和目的地址对在ACL为了控制流量配置的地址。您也可以将扩展 ACL 设置为采用更精细的控制粒度，将其配置为根据以下标准过滤数据流：

- 协议
- 端口号
- 差分服务代码点 (DSCP) 值
- 优先级值
- 同步序列号 (SYN) 位的状态

扩展 ACL 的命令语法格式为：

IP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
 {deny | permit} protocol source source-wildcard destination
 destination-wildcard
 [precedence precedence] [tos tos] [log | log-input]
 [time-range time-range-name][fragments]
```

Internet Control Message Protocol (ICMP)

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
 {deny | permit}
 icmp source source-wildcard destination destination-wildcard [icmp-type
 [icmp-code] | [icmp-message]] [precedenceprecedence] [tos tos] [log |
 log-input] [time-range time-range-name][fragments]
```

传输控制协议 (TCP)

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} tcp
  source source-wildcard [operator [port]] destination destination-wildcard
  [operator [port]] [established] [precedence precedence] [tos tos] [log |
  log-input] [time-range time-range-name][fragments]
```

用户数据报协议 (UDP)

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} udp
  source source-wildcard [operator [port]] destination destination-wildcard
  [operator [port]] [precedence precedence] [tos tos] [log | log-input]
  [time-range time-range-name][fragments]
```

先决条件

要求

在尝试进行此配置之前，请确保满足以下要求：

- 对 IP 编址有一定的基本了解

有关其他信息，请参阅 [IP 编址和子网划分入门指南](#)。

使用的组件

本文档不限于特定的软件和硬件版本。

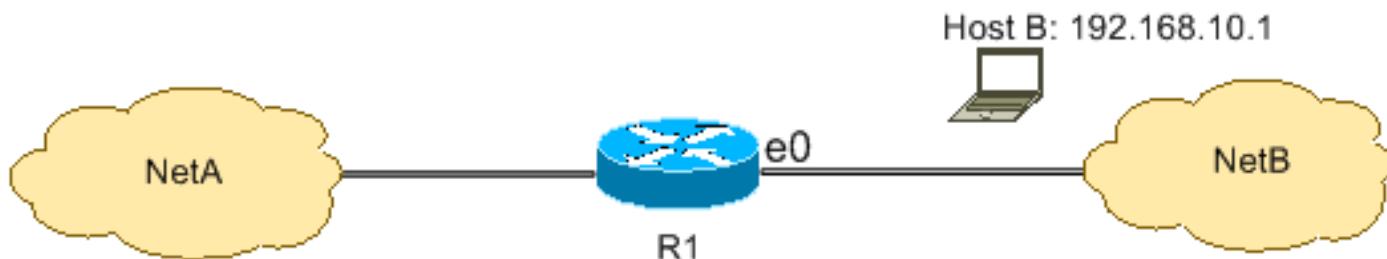
本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

配置

这些配置示例使用了最常见的 IP ACL。

允许选定主机访问网络

下图显示了向一台选定主机授予网络访问权限。从主机 B 发往 NetA 的所有数据流都得到允许，而其他所有从 NetB 发往 NetA 的数据流都遭到拒绝。



R1 表中的输出显示了网络如何向主机授予访问权限。该输出说明：

- 此配置只允许 IP 地址为 192.168.10.1 的主机上的数据流通过 R1 上的以太网 0 接口。
- 这台主机有权访问 NetA 的 IP 服务。

- NetB 中的其他主机无权访问 NetA。
- ACL 中没有配置 deny 语句。

默认情况下，每个 ACL 末尾都有一个隐式 deny all 子句。任何没有显式允许的数据流都将遭到拒绝。

R1

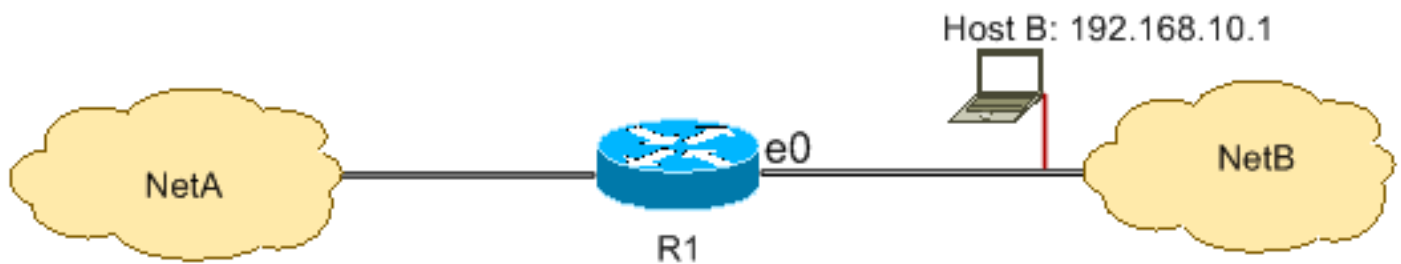
```
hostname R1
!
interface ethernet0
ip access-group 1 in
!
access-list 1 permit host 192.168.10.1
```

注意： ACL 过滤了从 NetB 传输到 NetA 的 IP 数据包，从主机 B 发出的数据包则除外。从主机 B 发出的数据包到 NetA 仍然允许。

注意： ACL `access-list 1 permit 192.168.10.1 0.0.0.0` 是配置相同规则的另一种方法。

拒绝选定主机访问网络

下图显示了这种情况：从主机 B 发往 NetA 的数据流遭到拒绝，而其他所有从 NetB 发出以访问 NetA 的数据流都得到了允许。



此配置拒绝主机 192.168.10.1/32 上的所有数据包通过 R1 上的以太网 0 接口，但允许其他所有数据包。由于每个 ACL 都有一个隐式 deny all 子句，因此您必须使用 `access list 1 permit any` 命令显式允许其他所有数据包。

R1

```
hostname R1
!
interface ethernet0
ip access-group 1 in
!
access-list 1 deny host 192.168.10.1
access-list 1 permit any
```

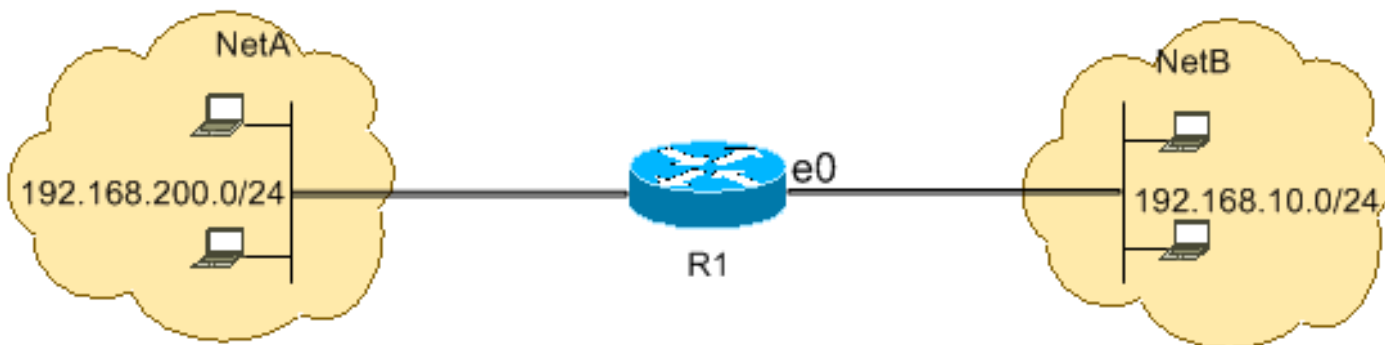
注意： 语句顺序对 ACL 的操作至关重要。如果颠倒了条目顺序，如此命令所示，则第一行与每个数据包的源地址都匹配。因此，ACL 无法阻止主机 192.168.10.1/32 访问 NetA。

```
access-list 1 permit any
access-list 1 deny host 192.168.10.1
```

允许访问一个范围内的连续 IP 地址

下图显示了 NetB 中网络地址为 192.168.10.0/24 的所有主机都能访问 NetA 中地址为

192.168.200.0/24 的网络。



如果某个 IP 数据包的 IP 报头显示的源地址在网络 192.168.10.0/24 中,而目标地址在网络 192.168.200.0/24 中,此配置将允许这种数据包访问 NetA。ACL 末尾有隐式 deny all 子句,将拒绝其他所有数据流通过 R1 上的以太网 0 入站。

R1

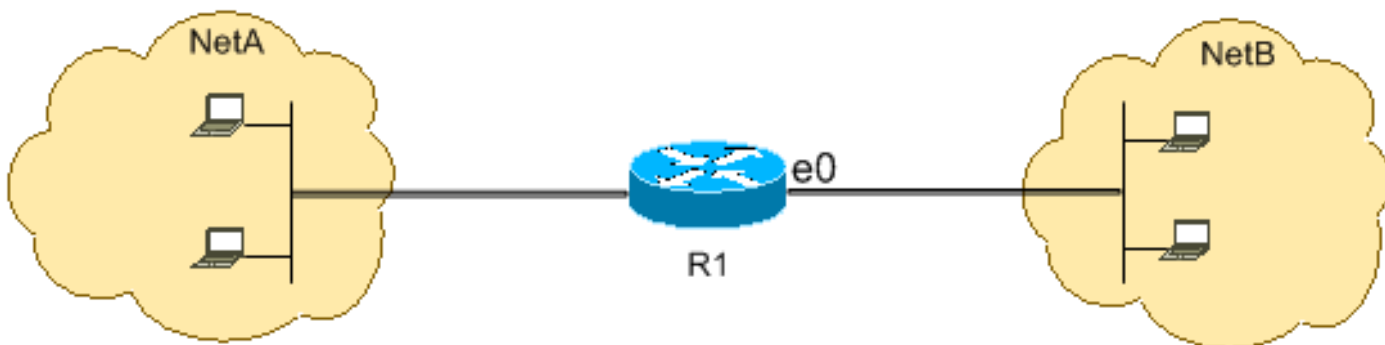
```
hostname R1
!
interface ethernet0
ip access-group 101 in
!
access-list 101 permit ip 192.168.10.0 0.0.0.255
192.168.200.0 0.0.0.255
```

注意：在 `access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.200.0 0.0.0.255` 命令中，“0.0.0.255”是掩码为 255.255.255.0 的网络 192.168.10.0 的反掩码。ACL 使用反掩码来确定网络地址中需要匹配的位数。上表中，ACL 允许源地址位于 192.168.10.0/24 网络中且目标地址位于 192.168.200.0/24 网络中的所有主机。

要详细了解网络地址的掩码以及如何计算 ACL 所需的反掩码，请参阅[配置 IP 访问列表的掩码部分](#)。

拒绝 Telnet 流量 (TCP、端口 23)

为了满足更高的安全要求，您可能需要禁止从公共网络对您的专用网络发起 Telnet 访问会话。下图显示了如何拒绝从 NetB (公共) 发往 NetA (专用) 的 Telnet 数据流，但同时允许其他所有 IP 数据流，也允许 NetA 发起并建立与 NetB 的 Telnet 会话。



Telnet 使用 TCP 端口 23。此配置显示了通过端口 23 发往 NetA 的所有 TCP 数据流都受到了阻止，但其他所有 IP 数据流都得到了允许。

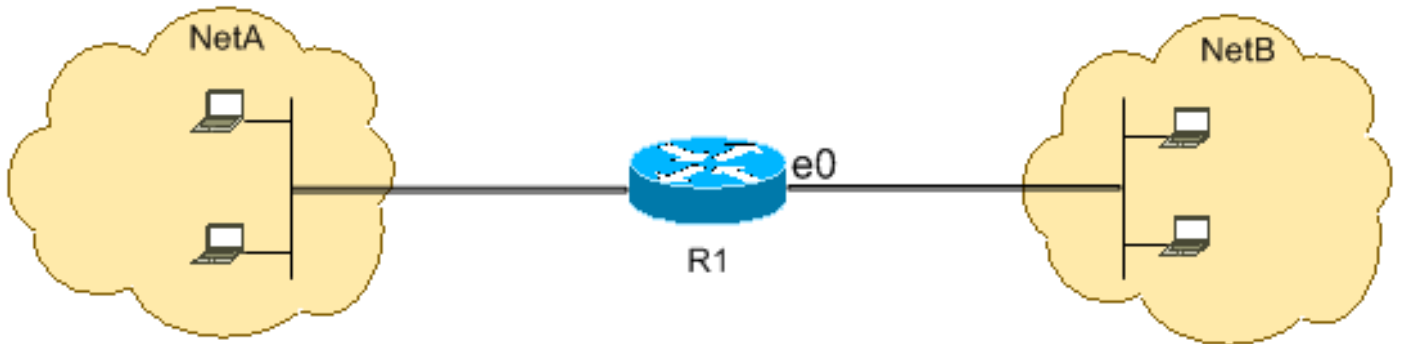
R1

```
hostname R1
```

```
!  
interface ethernet0  
ip access-group 102 in  
!  
access-list 102 deny tcp any any eq 23  
access-list 102 permit ip any any
```

只允许内部网络发起 TCP 会话

下图显示了允许从 NetA 发往 NetB 的 TCP 数据流，但拒绝从 NetB 发往 NetA 的 TCP 数据流。



本示例中 ACL 的目的是：

- 允许 NetA 中的主机面向 NetB 中的主机发起并建立 TCP 会话。
- 拒绝 NetB 中的主机面向 NetA 中的主机发起并建立 TCP 会话。

当数据报具备以下条件时，此配置允许数据报通过 R1 上的以太网 0 接口入站：

- 已设置确认 (ACK) 或重置 (RST) 位 (指示一个已建立的 TCP 会话)
- 目标端口值大于 1023

R1

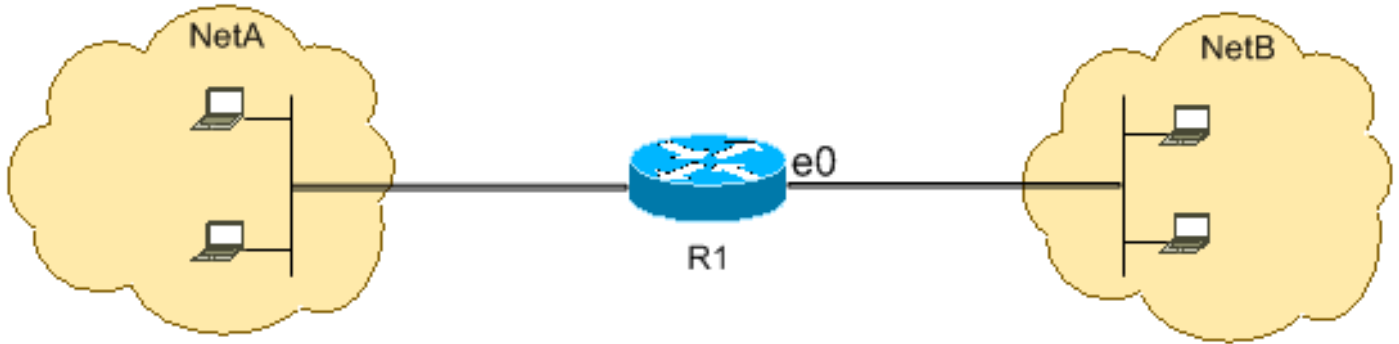
```
hostname R1  
!  
interface ethernet0  
ip access-group 102 in  
!  
access-list 102 permit tcp any any gt 1023 established
```

由于 IP 服务的大多数常用端口都使用小于 1023 的值，ACL 102 将拒绝目标端口值小于 1023 或未设置 ACK/RST 位的所有数据报。因此，当 NetB 中的一台主机在未设置同步/开始数据包 (SYN) 位的情况下面向一个编号小于 1023 的端口通过发送第一个 TCP 数据包来发起 TCP 连接时，它将遭到拒绝，并且 TCP 会话将失败。从 NetA 发往 NetB 的 TCP 会话将得到允许，因为它已设置用于返回数据包的 ACK/RST 位并且使用的端口值大于 1023。

有关端口的完整列表，请参阅 [RFC 1700](#)。

拒绝 FTP 数据流 (TCP、端口 21)

下图显示了拒绝从 NetB 发往 NetA 的 FTP (TCP，端口 21) 和 FTP 数据 (端口 20) 流量，但允许其他所有 IP 数据流。



FTP 使用端口 21 和端口 20。发往端口 21 和端口 20 的 TCP 数据流遭到拒绝，但其他所有数据流得到显式允许。

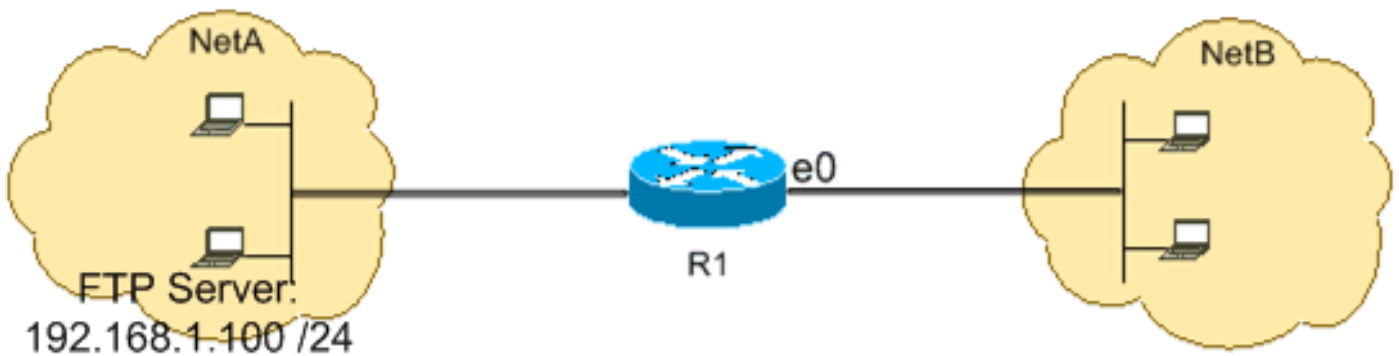
R1

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 deny tcp any any eq ftp
access-list 102 deny tcp any any eq ftp-data
access-list 102 permit ip any any
```

允许 FTP 数据流 (主动 FTP)

FTP 能以主动和被动两种不同模式进行操作。要了解主动和被动 FTP 的运作方式，请参阅 [FTP 操作](#)。

当 FTP 以主动模式操作时，FTP 服务器将端口 21 用于控制，将端口 20 用于数据。FTP 服务器 (192.168.1.100) 位于 NetA 中。下图显示了允许从 NetB 发往 FTP 服务器 (192.168.1.100) 的 FTP (TCP , 端口 21) 和 FTP 数据 (端口 20) 流量，但拒绝其他所有 IP 数据流。



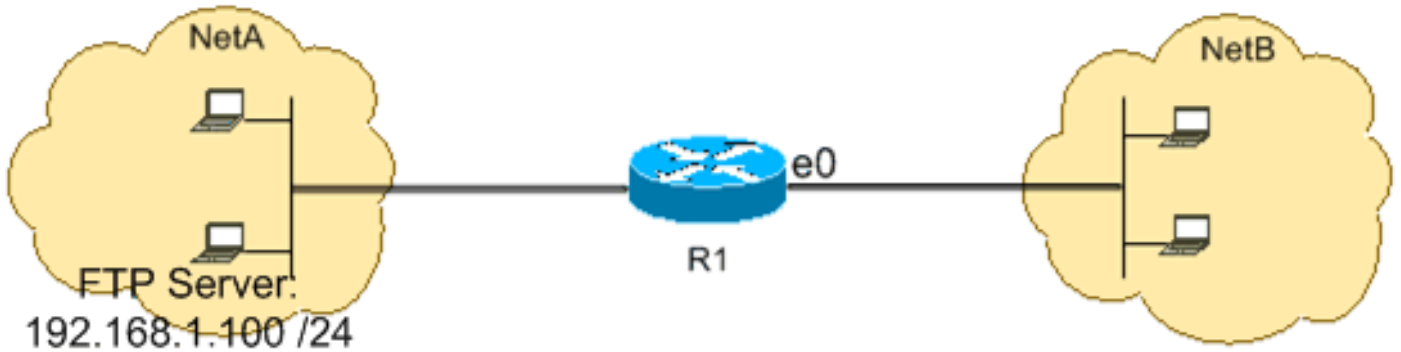
R1

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit tcp any host 192.168.1.100 eq ftp
access-list 102 permit tcp any host 192.168.1.100 eq ftp-data established
!
interface ethernet1
ip access-group 110 in
!
access-list 110 permit host 192.168.1.100 eq ftp any established
access-list 110 permit host 192.168.1.100 eq ftp-data any
```

允许 FTP 数据流 (被动 FTP)

FTP 能以主动和被动两种不同模式进行操作。要了解主动和被动 FTP 的运作方式，请参阅 [FTP 操作](#)。

当 FTP 以被动模式操作时，FTP 服务器将端口 21 用于控制，将编号大于或等于 1024 的动态端口用于数据。FTP 服务器 (192.168.1.100) 位于 NetA 中。下图显示了允许从 NetB 发往 FTP 服务器 (192.168.1.100) 的 FTP (TCP，端口21) 和 FTP 数据 (端口大于或等于 1024) 流量，但拒绝其他所有 IP 数据流。

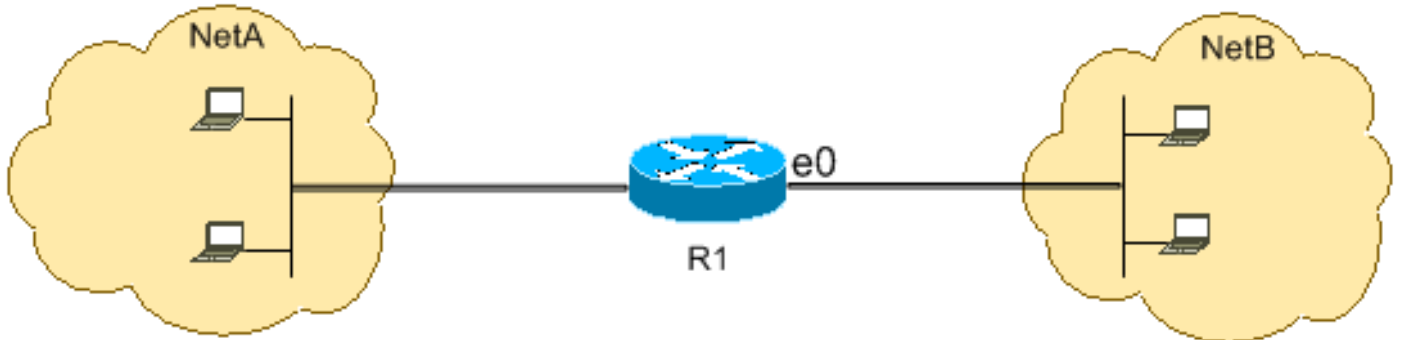


R1

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit tcp any host 192.168.1.100 eq ftp
access-list 102 permit tcp any host 192.168.1.100 gt 1024
!
interface ethernet1
ip access-group 110 in
!
access-list 110 permit host 192.168.1.100 eq ftp any established
access-list 110 permit host 192.168.1.100 gt 1024 any established
```

允许 ping (ICMP)

下图显示了允许从 NetA 发往 NetB 的 ICMP，但拒绝从 NetB 发往 NetA 的 ping。



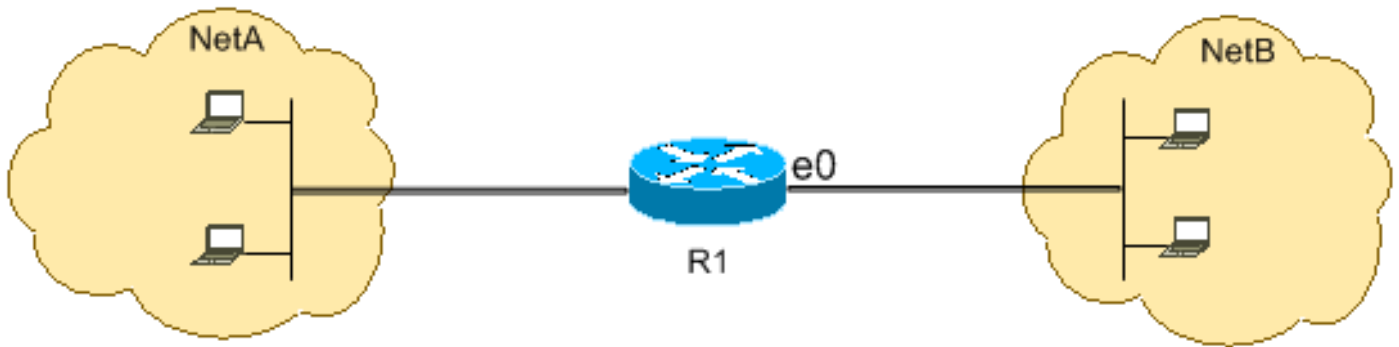
此配置仅允许从 NetB 到 NetA 的 Echo 应答 (ping 响应) 数据包通过以太网 0 接口入站。但是，当 ping 是从 NetB 发往 NetA 时，此配置将阻止所有的 Echo 请求 ICMP 数据包。因此，NetA 中的主机能对 NetB 中的主机执行 ping 操作，但 NetB 中的主机不能对 NetA 中的主机执行 ping 操作。

R1


```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit icmp any any echo-reply
```

允许 HTTP、Telnet、Mail、POP3、FTP

下图显示了仅允许 HTTP、Telnet、Simple Mail Transfer Protocol (SMTP)、POP3 和 FTP 数据流，并拒绝其他所有从 NetB 发往 NetA 的数据流。



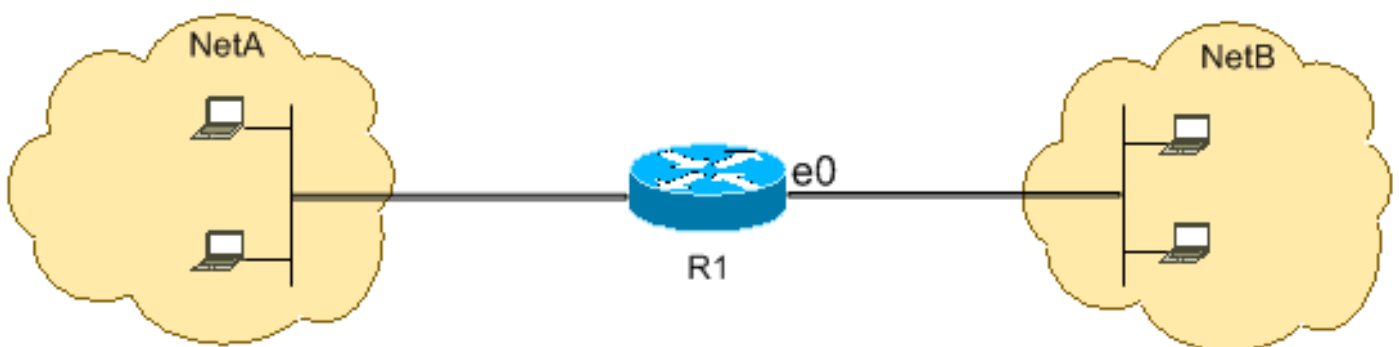
此配置允许目标端口值与 WWW (端口 80)、Telnet (端口 23)、SMTP (端口 25)、POP3 (端口 110)、FTP (端口 21) 或 FTP 数据 (端口 20) 匹配的 TCP 数据流。请注意，ACL 末尾的隐式 deny all 字句将拒绝其他所有不匹配 permit 子句的数据流。

R1

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq telnet
access-list 102 permit tcp any any eq smtp
access-list 102 permit tcp any any eq pop3
access-list 102 permit tcp any any eq 21
access-list 102 permit tcp any any eq 20
```

允许 DNS

下图显示了仅允许域名系统 (DNS) 数据流，并拒绝其他所有从 NetB 发往 NetA 的数据流。



此配置允许目标端口值为 53 的 TCP 数据流。ACL 末尾的隐式 deny all 子句将拒绝其他所有不匹配 permit 子句的数据流。

R1

```
hostname R1
!
interface ethernet0
ip access-group 102 in
!
access-list 112 permit udp any any eq domain
access-list 112 permit udp any eq domain any
access-list 112 permit tcp any any eq domain
access-list 112 permit tcp any eq domain any
```

允许路由更新

当您对接口应用入站 ACL 时，请确保路由更新没有被过滤掉。使用此列表中的相关 ACL，以便允许路由协议数据包：

输入此命令为了允许路由信息协议(RIP)：

```
access-list 102 permit udp any any eq rip
```

输入此命令为了允许增强型内部网关路由协议(EIGRP)：

```
access-list 102 permit igmp any any
```

输入此命令为了允许高级IGMP (EIGRP)：

```
access-list 102 permit eigrp any any
```

输入此命令为了允许开放最短路径优先(OSPF)：

```
access-list 102 permit ospf any any
```

输入此命令为了允许边界网关协议(BGP)：

```
access-list 102 permit tcp any any eq 179
```

```
access-list 102 permit tcp any eq 179 any
```

[根据 ACL 调试数据流](#)

要使用 **debug** 命令就需要分配系统资源（如内存和处理能力），在极少数情况下会导致系统因负载过高而停止运行。因此请慎用 **debug** 命令。请使用 ACL，以便有选择性地定义需要检查的数据流，从而减少 debug 命令的影响。这样的配置不会过滤任何数据包。

此配置仅针对主机 10.1.1.1 和主机 172.16.1.1 之间的数据包启用 **debug ip packet** 命令。

```
R1(config)#access-list 199 permit tcp host 10.1.1.1 host 172.16.1.1
R1(config)#access-list 199 permit tcp host 172.16.1.1 host 10.1.1.1
R1(config)#end
R1#debug ip packet 199 detail
IP packet debugging is on (detailed) for access list 199
```

要了解关于 debug 命令影响的其他信息，请参阅[有关 debug 命令的重要信息](#)。

要了解关于配合使用 ACL 与 debug 命令的其他信息，请参阅[了解 ping 和 traceroute 命令中的使用 debug 命令](#)部分。

Mac 地址过滤

您可以过滤带有特定 MAC 层站点的源地址或目标地址的帧。可以向系统中配置任意数量的地址，而不会对性能产生影响。要按照 MAC 层地址进行过滤，请在全局配置模式下使用以下命令：

```
Router#config terminal
    bridge irb
    bridge 1 protocol ieee
    bridge 1 route ip
```

将网桥协议和创建的访问列表应用到要过滤数据流的接口上：

```
Router#int fa0/0
    no ip address
    bridge-group 1 {input-address-list 700 | output-address-list 700}
    exit
```

创建桥接虚拟接口，并将分配的 IP 地址应用到以太网接口上：

```
Router#int bvi1
    ip address
    exit
!
!
    access-list 700 deny <mac address> 0000.0000.0000
    access-list 700 permit 0000.0000.0000 ffff.ffff.ffff
```

使用此配置后，路由器仅允许在访问列表 700 上配置的 MAC 地址。使用该访问列表将拒绝无访问权限的 MAC 地址，然后允许其余地址。

注意：为每个 MAC 地址创建访问列表的每一行条目。

[验证](#)

当前没有可用于此配置的验证过程。

[故障排除](#)

目前没有针对此配置的故障排除信息。

[相关信息](#)

- [配置 IP 访问列表](#)
- [访问列表支持页面](#)
- [IP 路由支持页](#)
- [IP 路由协议支持页](#)
- [技术支持和文档 - Cisco Systems](#)