# Configure Commonly Used IP ACLs

## Contents


**Introduction**

**Prerequisites**

- Requirements
- Components Used

**Background Information**

**Configure**

- Allow a Select Host to Access the Network
- Deny a Select Host to Access the Network
- Allow Access to a Range of Contiguous IP Addresses
- Deny Telnet Traffic (TCP, Port 23)
- Allow Only Internal Networks to Initiate a TCP Session
- Deny FTP Traffic (TCP, Port 21)
- Allow FTP Traffic (Active FTP)
- Allow FTP Traffic (Passive FTP)
- Allow Pings (ICMP)
- Allow HTTP, Telnet, Mail, POP3, FTP
- Allow DNS
- Permit Routing Updates
- Debug Traffic Based on ACL
- MAC Address Filtering

**Verify**

**Troubleshoot**

**Related Information**


## Introduction

This document describes sample configurations for commonly used IP Access Control Lists (ACLs), which filter IP packets.

## Prerequisites

### Requirements

Ensure that you meet this requirement before you attempt this configuration:

- Basic understanding of IP addressing.

Refer to IP Addressing and Subnetting for New Users for additional information.

### Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

# Background Information

While this document discusses some commonly used standard and extended ACLs, for more detailed information on different types of ACLs supported by Cisco IOS software, as well as ow to configure and edit ACLs, refer to Configuring IP Access Lists.

IP Access Control Lists filter packets based on:

- Source address
- Destination address
- Type of packet
- Any combination of these items

In order to filter network traffic, ACLs control whether routed packets are forwarded or blocked at the router interface. Your router examines each packet in order to determine whether to forward or drop the packet based on the criteria that you specify within the ACL. ACL criteria include:

- Source address of the traffic
- Destination address of the traffic
- Upper-layer protocol

Complete these steps in order to construct an ACL as the examples in this document show:

1. Create an ACL.
2. Apply the ACL to an interface.

The IP ACL is a sequential collection of permit and deny conditions that apply to an IP packet. The router tests packets against the conditions in the ACL one at a time.

The first match determines whether the Cisco IOS® Software accepts or rejects the packet. Because the Cisco IOS Software stops the test of conditions after the first match, the order of the conditions is critical. If no conditions match, the router rejects the packet because of an implicit deny all clause.

These are examples of IP ACLs that can be configured in Cisco IOS Software:

- Standard ACLs
- Extended ACLs
- Dynamic (lock and key) ACLs
- IP-named ACLs
- Reflexive ACLs
- Time-based ACLs that use time ranges
- Commented IP ACL entries
- Context-based ACLs
- Authentication proxy
- Turbo ACLs
- Distributed time-based ACLs

The command syntax format of a standard ACL is **access-list access-list-number {permit|deny} {host|source source-wildcard|any}**.

Standard ACLs compare the source address of the IP packets to the addresses configured in the ACL in order to control traffic.

Extended ACLs compare the source and destination addresses of the IP packets to the addresses configured in the ACL in order to control traffic. You can also make extended ACLs more granular and configured to filter traffic by criteria such as:

- Protocol
- Port numbers
- Differentiated services code point (DSCP) value
- Precedence value
- State of the synchronize sequence number (SYN) bit

The command syntax formats of extended ACLs are:

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]] [precedence precedence] [tos tos] [log | log-input]
 [time-range time-range-name][fragments]
```

Transport Control Protocol (TCP)

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

User Datagram Protocol (UDP)

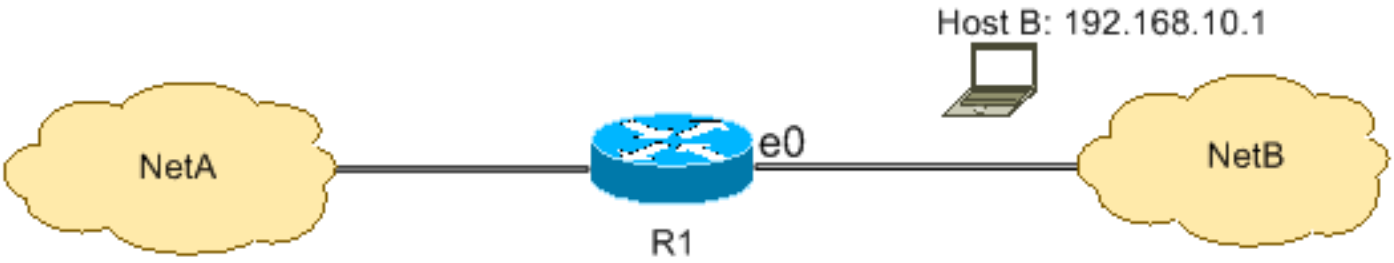
```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
{deny | permit} udp source source-wildcard [operator [port]] destination destination-wildcard [operator
 [precedence precedence] [tos tos] [log | log-input]
 [time-range time-range-name][fragments]
```

# Configure

These configuration examples use the most common IP ACLs.

## Allow a Select Host to Access the Network

This image shows that a select host is granted permission to access the network. All traffic sourced from Host B destined to NetA is permitted, and all other traffic sourced from NetB destined to NetA is denied.




The output on the R1 table shows how the network grants access to the host. This output shows that:

- The configuration allows only the host with the IP address 192.168.10.1 through the Ethernet 0 interface on R1.
- This host has access to the IP services of NetA.
- No other host in NetB has access to NetA.
- No deny statement is configured in the ACL.

By default, there is an implicit deny all clause at the end of every ACL. Anything that is not explicitly permitted is denied.

R1

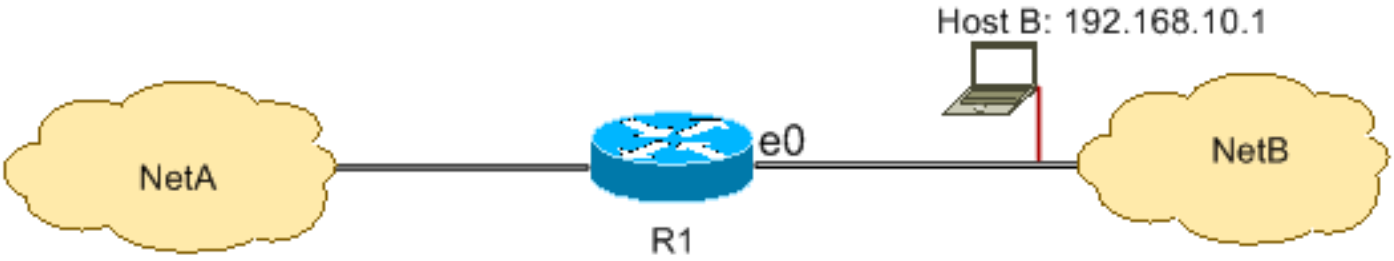
```
hostname R1
!
interface ethernet0
 ip access-group 1 in
!
access-list 1 permit host 192.168.10.1
```

**Note**: The ACL filters IP packets from NetB to NetA, except packets sourced from Host B. Packets sourced from Host B to NetA are still permitted.

**Note**: The ACL access-list 1 permit 192.168.10.1 0.0.0.0 is another way to configure the same rule.

## Deny a Select Host to Access the Network

This image shows that traffic sourced from Host B destined to NetA is denied, while all other traffic from the NetB to access NetA is permitted.




This configuration denies all packets from host 192.168.10.1/32 through Ethernet 0 on R1 and permits everything else. You must use the **access list 1 permit any** command to explicitly permit everything else because there is an implicit deny all clause with every ACL.

R1

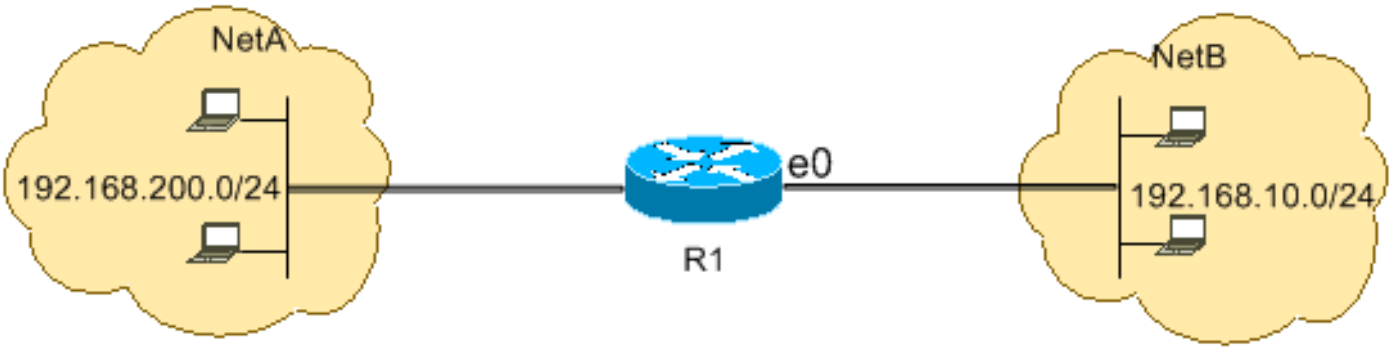
```
hostname R1
!
interface ethernet0
 ip access-group 1 in
!
access-list 1 deny host 192.168.10.1
access-list 1 permit any
```

**Note**: The order of statements is critical to the operation of an ACL. If the order of the entries is reversed as this command shows, the first line matches every packet source address. Therefore, the ACL fails to block host 192.168.10.1/32 from accessing NetA.

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

## Allow Access to a Range of Contiguous IP Addresses

This image shows that all hosts in NetB with the network address 192.168.10.0/24 can access network 192.168.200.0/24 in NetA.




This configuration allows the IP packets with an IP header that has a source address in the network 192.168.10.0/24 and a destination address in the network 192.168.200.0/24 access to NetA. There is the implicit deny all clause at the end of the ACL which denies all other traffic passage through Ethernet 0

inbound on R1.

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
```

---

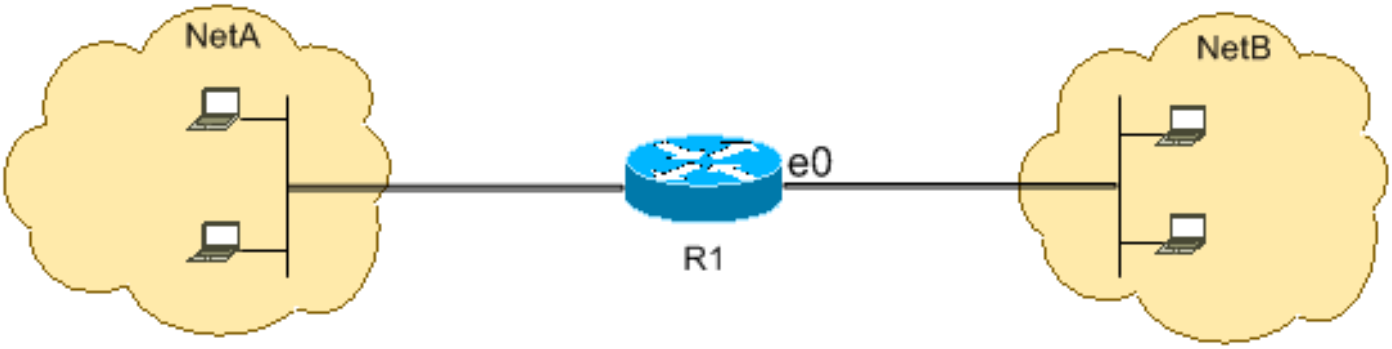
**Note**: In the command **access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.200.0 0.0.0.255**, the 0.0.0.255 is the inverse mask of network 192.168.10.0 with mask 255.255.255.0. ACLs use the inverse mask to know how many bits in the network address need to match. In the table, the ACL permits all hosts with source addresses in the 192.168.10.0/24 network and destination addresses in the 192.168.200.0/24 network.

---

Refer to the Masks section of Configuring IP Access Lists for more information on the mask of a network address and how to calculate the inverse mask needed for ACLs.

## Deny Telnet Traffic (TCP, Port 23)

In order to meet higher security concerns, you can disable Telnet access to your private network from the public network. This image shows how Telnet traffic from NetB (public) destined to NetA (private) is denied, which permits NetA to initiate and establish a Telnet session with NetB while all other IP traffic is permitted.



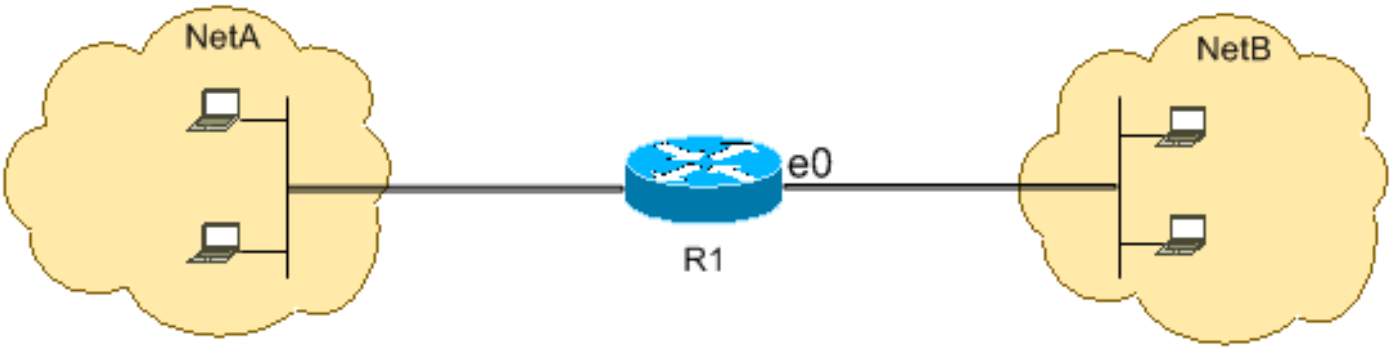

Telnet uses TCP, port 23. This configuration shows that all TCP traffic destined to NetA for port 23 is blocked, and all other IP traffic is permitted.

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
```

## Allow Only Internal Networks to Initiate a TCP Session

This figure shows that TCP traffic sourced from NetA destined to NetB is permitted, while TCP traffic from NetB destined to NetA is denied.




The purpose of the ACL in this example is to:

- Allow hosts in NetA to initiate and establish a TCP session to hosts in NetB.
- Deny hosts in NetB from initiating and establishing a TCP session destined to hosts in NetA.

This configuration allows a datagram to pass through interface Ethernet 0 inbound on R1 when the datagram has:

- Acknowledged (ACK) or reset (RST) bits set (indicates an established TCP session)
- A destination port value greater than 1023

R1

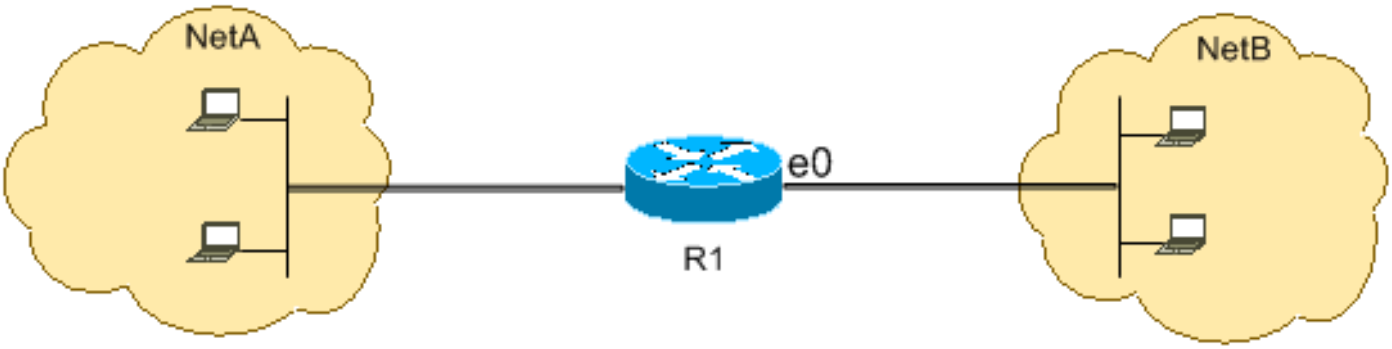
```
hostname R1
!
interface ethernet0
 ip access-group 102 in
!
access-list 102 permit tcp any any gt 1023 established
```

Since most of the well-known ports for IP services use values less than 1023, any datagram with a destination port less than 1023 or an ACK/RST bit not set is denied by ACL 102. Therefore, when a host from NetB initiates a TCP connection and sends the first TCP packet (without synchronize/start packet (SYN/RST) bit set) for a port number less than 1023, it is denied and the TCP session fails. The TCP sessions initiated from NetA destined to NetB are permitted because they have ACK/RST bit set for returning packets and use port values greater than 1023.

Refer to RFC 1700 for a complete list of ports.

## Deny FTP Traffic (TCP, Port 21)

This image shows that FTP (TCP, port 21) and FTP data (port 20 ) traffic sourced from NetB destined to NetA is denied, while all other IP traffic is permitted.




FTP uses port 21 and port 20. TCP traffic destined to port 21 and port 20 is denied and everything else is explicitly permitted.

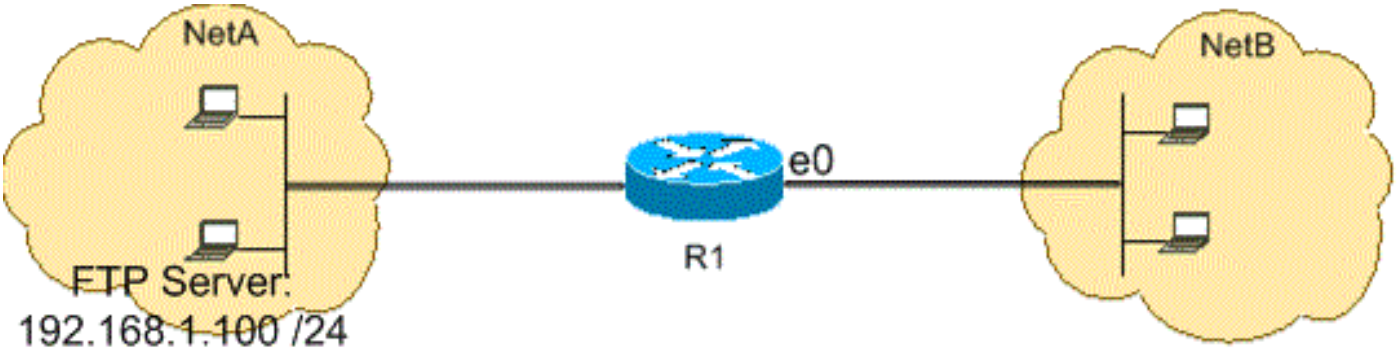
R1

```
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
```

## Allow FTP Traffic (Active FTP)

FTP can operate in two different modes named active and passive.

When FTP operates in active mode, the FTP server uses port 21 for control and port 20 for data. FTP server (192.168.1.100) is located in NetA. This image shows that FTP (TCP, port 21) and FTP data (port 20 ) traffic sourced from NetB destined to FTP server (192.168.1.100) is permitted, while all other IP traffic is denied.




R1

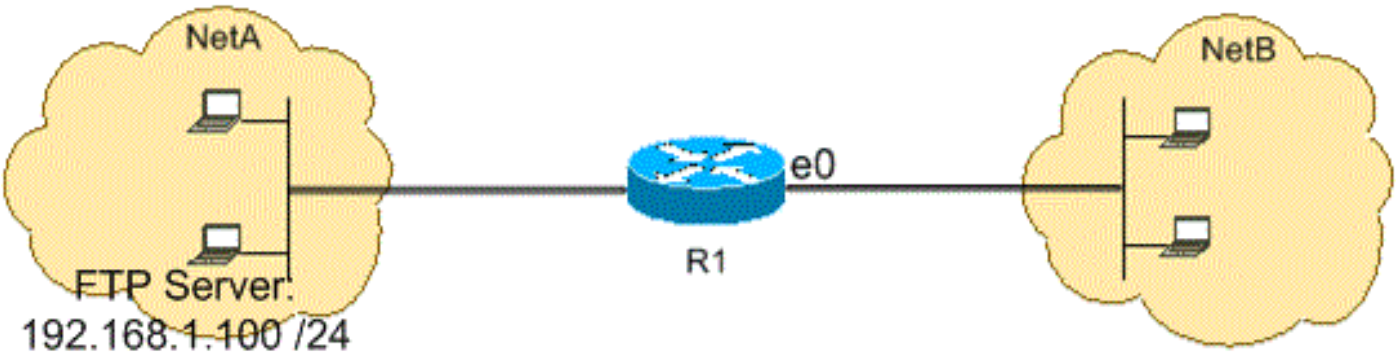
```
hostname R1
!
interface ethernet0
 ip access-group 102 in
!
```

```
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
```

## Allow FTP Traffic (Passive FTP)

FTP can operate in two different modes named active and passive.

When FTP operates in passive mode, the FTP server uses port 21 for control and the dynamic ports greater than or equal to 1024 for data. FTP server (192.168.1.100) is located in NetA. This image shows that FTP (TCP, port 21) and FTP data (ports greater than or equal to 1024) traffic sourced from NetB destined to FTP server (192.168.1.100) is permitted, while all other IP traffic is denied.

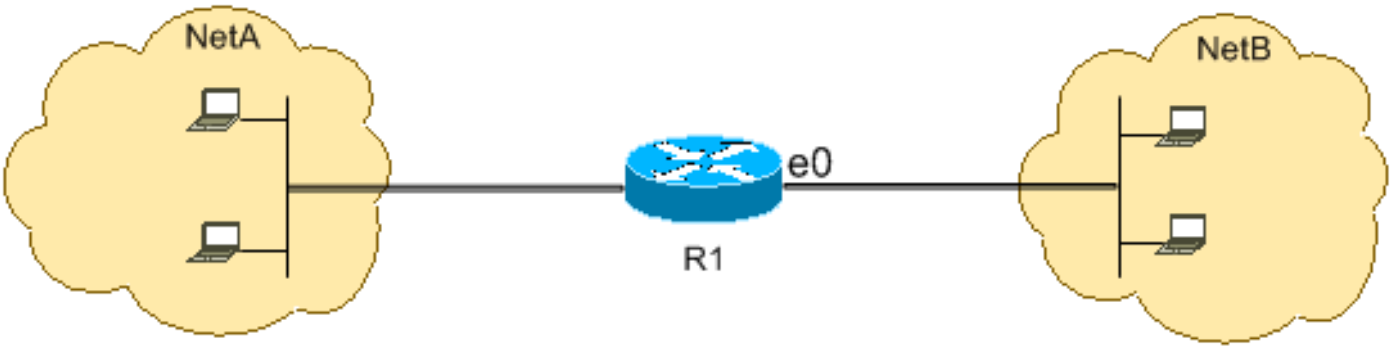



R1

```
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 1023
!
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 1023 any established
```

## Allow Pings (ICMP)

This image shows that ICMP sourced from NetA destined to NetB is permitted, and pings sourced from NetB destined to NetA are denied.



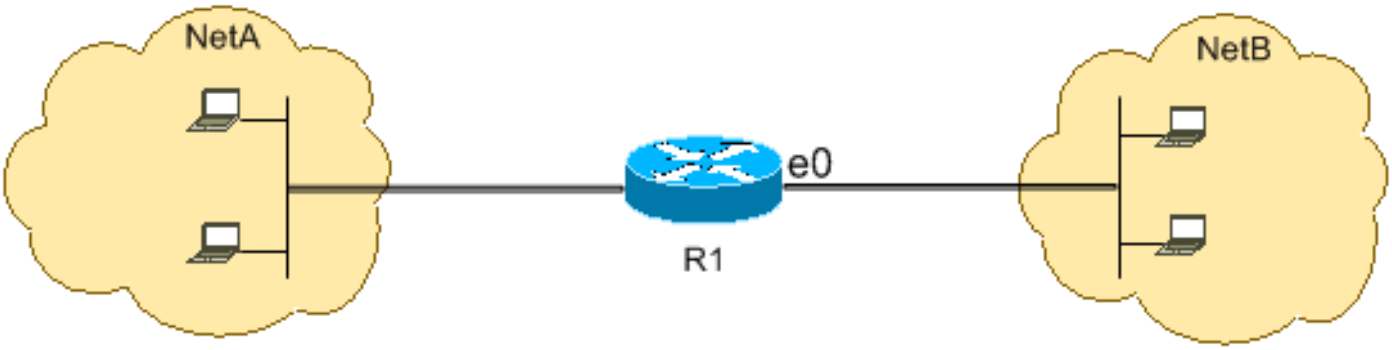

This configuration permits only echo-reply (ping response) packets to come in on interface Ethernet 0 from NetB towards NetA. However, the configuration blocks all echo-request ICMP packets when pings are sourced in NetB and destined to NetA. Therefore, hosts in NetA can ping hosts in NetB, but hosts in NetB cannot ping hosts in NetA.

R1

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

## Allow HTTP, Telnet, Mail, POP3, FTP

This image shows that only HTTP, Telnet, Simple Mail Transfer Protocol (SMTP), POP3, and FTP traffic are permitted, and the rest of the traffic sourced from NetB destined to NetA is denied.




This configuration permits TCP traffic with destination port values that match WWW (port 80), Telnet (port 23), SMTP (port 25), POP3 (port 110), FTP (port 21), or FTP data (port 20). Notice an implicit deny all clause at the end of an ACL denies all other traffic, which does not match the permit clauses.

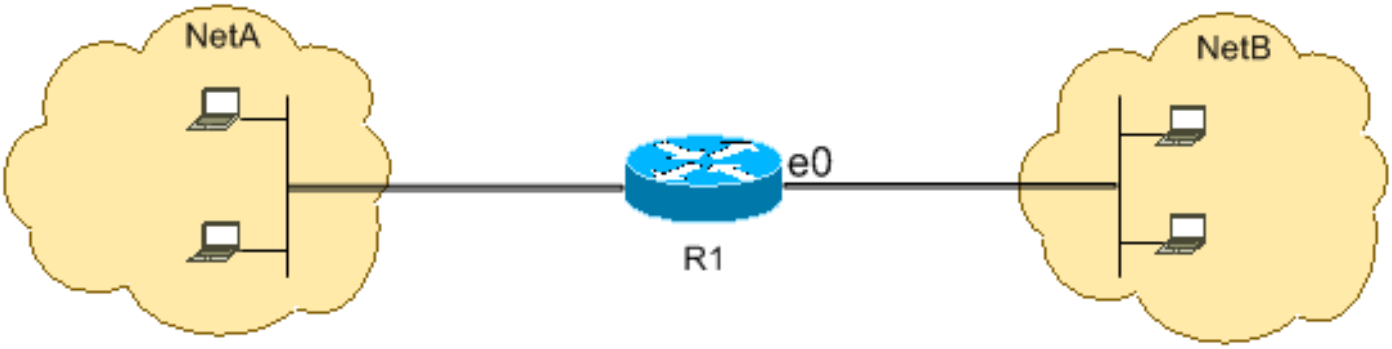
R1

```
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
```

## Allow DNS

This image shows that only Domain Name System (DNS) traffic is permitted, and the rest of the traffic sourced from NetB destined to NetA is denied.




This configuration permits TCP traffic with destination port value 53. The implicit deny all clause at the end of an ACL denies all other traffic, which does not match the permit clauses.

R1

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

## Permit Routing Updates

When you apply an in-bound ACL on to an interface, ensure that routing updates are not filtered out. Use the relevant ACL from this list to permit routing protocol packets:

Enter this command in order to permit Routing Information Protocol (RIP):

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

Enter this command in order to permit Interior Gateway Routing Protocol (IGRP):

```
access-list 102 permit igrp any any
```

Enter this command in order to permit Enhanced IGRP (EIGRP):

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

Enter this command in order to permit Open Shortest Path First (OSPF):

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

Enter this command in order to permit Border Gateway Protocol (BGP):

<#root>

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

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

## Debug Traffic Based on ACL

The use of **debug** commands requires the allocation of system resources like memory and processing power and in extreme situations can cause a heavily-loaded system to stall. Use **debug** commands with care. Use an ACL in order to selectively define the traffic that needs to be examined to reduce the impact of the **debug** command. Such a configuration does not filter any packets.

This configuration turns on the **debug ip packet** command only for packets between the hosts 10.1.1.1 and 172.16.1.1.

<#root>

```
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
```

Refer to Important Information on Debug Commands for additional information on the impact of debug commands.

Refer to the Use the Debug Command section of Understanding the Ping and Traceroute Commands for additional information on the use of ACLs with **debug** commands.

## MAC Address Filtering

You can filter frames with a particular MAC-layer station source or destination address. Any number of addresses can be configured into the system without a performance penalty. In order to filter by MAC-layer address, use this command in global configuration mode:

<#root>

Router#

**config terminal**

Router(config)#

**bridge irb**

Router(config)#

**bridge 1 protocol ieee**

Router(config)#

**bridge 1 route ip**

Apply the bridge protocol to an interface that you need to filter traffic along with the access list created with the **bridge-group <group number> {input-address-list <ACL number> | output-address-list <ACL number>}** command:

<#root>

Router#

**config terminal**

Router(config-if)#

**interface fastEthernet0/0**

Router(config-if)#

**no ip address**

Router(config-if)#

**bridge-group 1 input-address-list 700**

Router(config-if)#

**exit**

Create a Bridged Virtual Interface and apply the IP address that was assigned to the physical Ethernet interface:

<#root>

Router#

**config terminal**

Router(config-if)#

**int bvi1**

Router(config-if)#

**ip address 192.168.1.1 255.255.255.0**

Router(config-if)#

**exit**

Router(config)#

**access-list 700 deny aaaa.bbbb.cccc 0000.0000.0000**

Router(config)#

**access-list 700 permit 0000.0000.0000 ffff.ffff.ffff**

With this configuration, the router only allows the MAC addresses configured on the access-list 700. With the access list **access-list <ACL number> deny <mac address> 0000.0000.0000** command, deny the MAC address that cannot have access and then permit the rest (for this example, aaaa.bbbb.cccc).

---

**Note**: Create every line of access list for each MAC address.

---

# Verify

There is currently no verification procedure available for this configuration.

# Troubleshoot

There is currently no specific information available to troubleshoot this configuration.

## Related Information

- Configuring IP Access Lists
- Access Lists Support Page
- IP Routing Support Page
- IP Routed Protocols Support Page
- Technical Support & Documentation - Cisco Systems