Table of Contents

Configuring Commonly Used IP ACLs....1
Introduction....1
Prerequisites....2
Hardware and Software Versions....3
Configuration Examples....3
Allow a Select Host to Access the Network....3
Allow Access to a Range of Contiguous IP Addresses....3
Deny a Select Host to Access the Network....4
Deny Telnet Traffic (TCP, Port 23)....5
Allow Only Internal Networks to Initiate a TCP Session....5
Deny FTP Traffic (TCP, Port 21)....6
Allow Pings (ICMP)....6
Allow HTTP, Telnet, Mail, POP3, FTP....7
Permit Routing Updates....7
Related Information....8

Configuring Commonly Used IP ACLs

Introduction

Prerequisites
Hardware and Software Versions
Configuration Examples

Allow a Select Host to Access the Network
Allow Access to a Range of Contiguous IP Addresses
Deny a Select Host to Access the Network
Deny Telnet Traffic (TCP, Port 23)
Allow Only Internal Networks to Initiate a TCP Session
Deny FTP Traffic (TCP, Port 21)
Allow Pings (ICMP)
Allow HTTP, Telnet, Mail, POP3, FTP
Permit Routing Updates **Related Information**

Introduction

This document provides sample configurations of commonly used IP access control lists (ACL) which are commonly used to filter IP packets based on source addresses, destination addresses, type of packets, or any combination of those items.

ACLs filter network traffic by controlling whether routed packets are forwarded or blocked at the router interface. Your router examines each packet to determine whether to forward or drop the packet, based on the criteria you specified within the ACL. ACL criteria can be the source address of the traffic, the destination address of the traffic, or the upper−layer protocol.

The examples in this document show that an ACL is constructed using the following two steps:

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 address. The router tests addresses against the conditions in the ACL one at a time. The first match determines whether the Cisco IOS® software accepts or rejects the address. Because the Cisco IOS software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the router rejects the address, due to an implicit deny all clause.

There are many types of IP ACLs that can be configured in Cisco IOS, such as:

- Standard ACLs
- Extended ACLs
- Lock and Key (dynamic ACLs)
- IP named ACLs
- Reflexive ACLs
- Time−based ACLs using time ranges
- Commented IP ACL entries

- Context−based ACL
- Authentication proxy
- Turbo ACLs
- Distributed time−based ACLs

This document discusses some commonly used standard and extended ACLs. To understand more about the different types of ACLs supported in Cisco IOS software and to learn how to configure and edit ACLs, refer to Configuring IP Access Lists.

The syntax of a standard access−list command is shown below. Standard ACLs control traffic by comparing the source address of the IP packets to the addresses configured in the ACL.

```
access−list access−list−number {permit|deny} {host|source source−wildcard|any}
```

Extended ACLs control traffic by comparing the source and destination addresses of the IP packets to the addresses configured in the ACL.

The following is the command syntax format of extended ACLs.

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]
```

ICMP

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

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]
```

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]
```

The command reference for an ACL is available in IP Services Commands.

Prerequisites

Readers of this document should have a basic understanding of IP addressing. For additional information, refer to IP Addressing and Subnetting for New Users.

Hardware and Software Versions

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

Note: The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

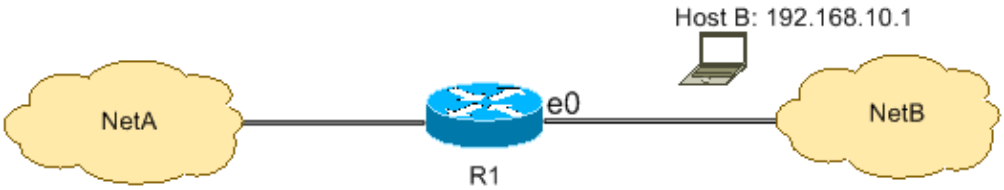
Configuration Examples

The following configuration examples use the most common IP ACLs.

Note: To find additional information on the commands used in this document, use the IOS Command Lookup Tool.

Allow a Select Host to Access the Network

The following is an example of a select host being granted permission to access the network. All traffic sourced from Host B destined to NetA is allowed, and all other traffic sourced from NetB destined to NetA is denied.



The following output shows how the host was granted access to the network. The configuration allows only the host with the IP address 192.168.10.1/32 through the Ethernet 0 interface on R1. This host will have access to the IP services of NetA. No other host in NetB will have access to NetA. There is no deny statement in the ACL . By default there is an implicit deny all at the end of every ACL. Anything that is not explicitly permitted is denied.

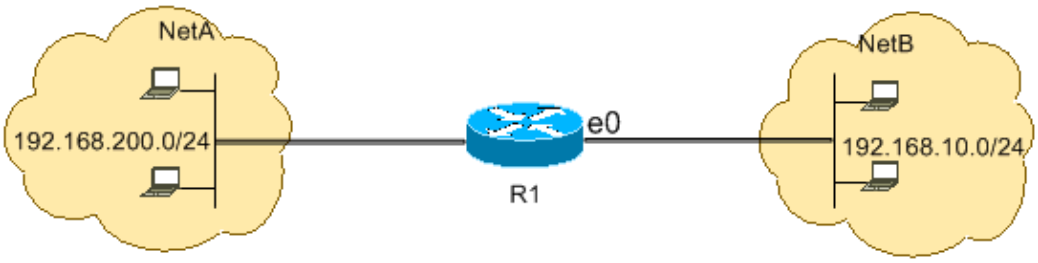
R1

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

Note: The above ACL filters IP packets sourced from Host B. Packets destined to Host B from NetA are still permitted.

Allow Access to a Range of Contiguous IP Addresses

The following example shows that all hosts in NetB with the network address 192.168.10.0/24 are allowed to access network 192.168.200.0/24 in NetA.

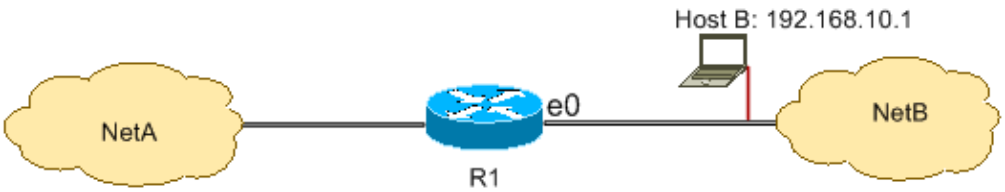


The following configuration allows the IP packets whose IP header has source network number of192.168.10.0/24 and a destination network number of 192.168.200.0/24 access to NetA. Again, there is the implicit deny all clause at the end of the ACL which denies all other traffic passage through Ethernet 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

Deny a Select Host to Access the Network

The following example shows that traffic sourced from Host B destined to NetA being denied, while permitting all other traffic from the NetB to access NetA.



The following configuration denies all packets from host 192.168.10.1/32 through Ethernet 0 on R1 and permits everything else. Because there is an implicit deny all clause with every ACL, you must explicitly permit everything else by using the **access list 1 permit any** command.

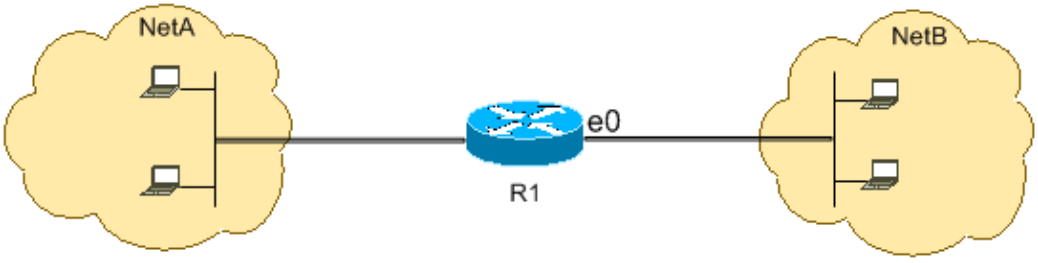
R1
hostname R1 ! interface ethernet0 ip access-group 1 in ! access-list 1 deny host 192.168.10.1 access-list 1 permit any

Note: The order of statements is critical to the operation of an ACL. If the order of entries is reversed as shown below, the first line would match every packet source address. Therefore, the ACL would fail 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
```

Deny Telnet Traffic (TCP, Port 23)

Disabling Telnet access to your private network from public network may be required to meet higher security concerns. The following example shows how Telnet traffic from NetB (public) destined to NetA (private) is denied, permitting NetA to initiate and establish Telnet session with NetB while permitting all other IP traffic.

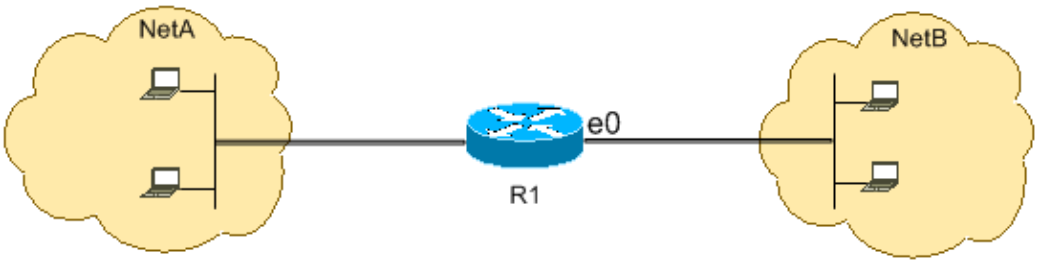


Telnet uses TCP, Port 23. The following configuration shows that all TCP traffic destined to NetA for Port 23 is blocked, and all other IP traffic is permitted.

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

The following example shows TCP traffic sourced from NetA destined to NetB being allowed, while denying TCP traffic from NetB destined to NetA.



The purpose of the ACL in this example is to achieve the following:

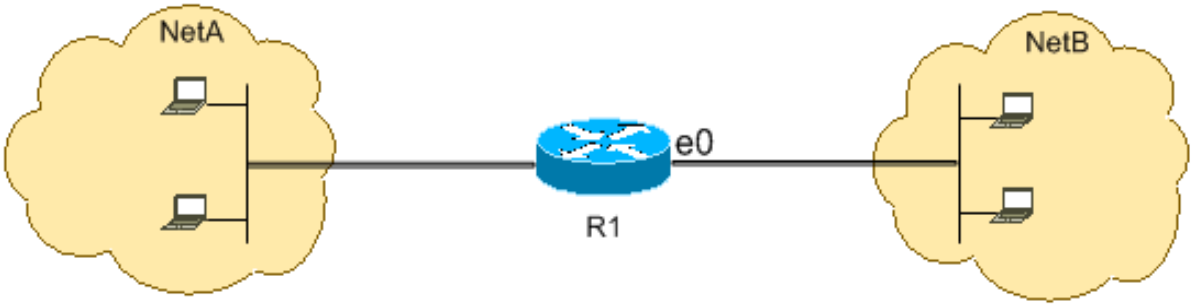
- Allow hosts in NetA to initiate and establish a TCP session to hosts in NetB.
- Deny hosts in NetB from initiating and establishing a TCP session destined to hosts in NetA.

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

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 (indicating an established TCP session), and has a destination port value greater than 1023. Since most of the well−known ports for IP services uses values below 1023, any datagram with a destination port less than 1023 and/or an ACK/RST bit not set, will be denied by ACL 102. Therefore, when a host from NetB initiates a TCP connection by sending the first TCP packet (without synchronize/start packet (SYN/RST) bit set) for a port number less than 1023, it will be denied and the TCP session will fail. The TCP sessions initiated from NetA destined to NetB will be permitted because they will have ACK/RST bit set for returning packets and will be using port values greater than 1023. For a complete list of ports, refer to RFC 1700.

Deny FTP Traffic (TCP, Port 21)

The following example shows FTP (TCP, port 21) and FTP−Data (port 20) traffic sourced from NetB destined to NetA being denied, while permitting all other IP traffic.



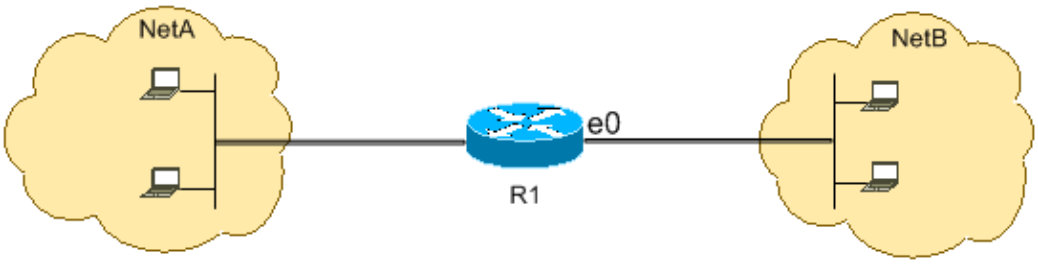
FTP uses Port 21 and Port 20. So, TCP traffic destined to port 21 and Port 20 is denied and everything else is explicitly permitted.

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 Pings (ICMP)

The following example shows Internet Control Message Protocol (ICMP) originating from NetA destined to NetB being allowed, and denying any pings originating from NetB destined to NetA.



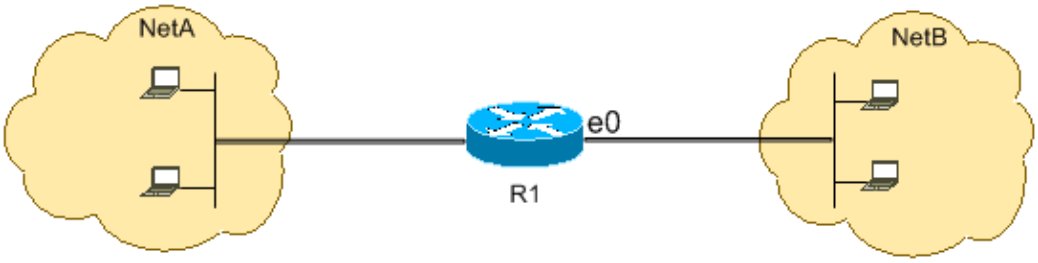
The following configuration, permits only echo−reply (ping response) packets to come in on interface Ethernet 0 from NetB toward NetA. However, the configuration blocks all echo−request ICMP packets when

pings are originated in NetB destined to NetA. Therefore, hosts in NetA can ping hosts in NetB but hosts in Net B 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

The following example shows only HTTP, Telnet, Simple Mail Transfer Protocol (SMTP), POP3, and FTP traffic being allowed, and denies the rest of the traffic sourced from NetB destined to NetA. Refer to the diagram above.



The following configuration permits TCP traffic with destination port values matching WWW (port 80), Telnet (port 23), SMTP (port 25), POP3 (port 110), or FTP (port 21). 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 pop3 access−list 102 permit tcp any any eq 21

Permit Routing Updates

Whenever you are applying an in−bound ACL on an interface, ensure routing updates are not filtered out. To permit routing protocol packets, use the relevant ACL below to ensure routing updates are not filtered out.

To permit Routing Information Protocol (RIP) use:

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

To permit Interior Gateway Routing Protocol (IGRP) use:

```
access−list 102 permit igrp any any
```

To permit Enhanced IGRP (EIGRP) use:

```
access−list 102 permit eigrp any any
```

To permit Open Shortest Path First (OSPF) use:

```
access−list 102 permit ospf any any
```

To permit Border Gateway Protocol (BGP) use:

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

Related Information

- **TCP/IP Routing and Routed Protocols Support Page**