

# Protecting Your Core: Infrastructure Protection Access Control Lists

Document ID: 43920

---

## Introduction

### Infrastructure Protection

Background

Techniques

### ACL Examples

#### Develop a Protection ACL

#### ACLs and Fragmented Packets

#### Risk Assessment

### Appendices

Supported IP Protocols in Cisco IOS Software

Deployment Guidelines

Deployment Examples

### NetPro Discussion Forums – Featured Conversations

### Related Information

---

## Introduction

This document presents guidelines and recommended deployment techniques for infrastructure protection access control lists (ACLs). Infrastructure ACLs are used to minimize the risk and effectiveness of direct infrastructure attack by explicitly permitting only authorized traffic to the infrastructure equipment while permitting all other transit traffic.

## Infrastructure Protection

### Background

In an effort to protect routers from various risks both accidental and malicious infrastructure protection ACLs should be deployed at network ingress points. These IPv4 and IPv6 ACLs deny access from external sources to all infrastructure addresses, such as router interfaces. At the same time, the ACLs permit routine transit traffic to flow uninterrupted and provide basic RFC 1918 , RFC 3330 , and anti-spoof filtering.

Data received by a router can be divided into two broad categories:

- traffic that passes through the router via the forwarding path
- traffic destined for the router via the receive path for route processor handling

In normal operations, the vast majority of traffic simply flows through a router en route to its ultimate destination.

However, the route processor (RP) must handle certain types of data directly, most notably routing protocols, remote router access (such as Secure Shell [SSH]), and network management traffic such as Simple Network Management Protocol (SNMP). In addition, protocols such as Internet Control Message Protocol (ICMP) and IP options can require direct processing by the RP. Most often, direct infrastructure router access is required only from internal sources. A few notable exceptions include external Border Gateway Protocol (BGP) peering, protocols that terminate on the actual router (such as generic routing encapsulation [GRE] or IPv6

over IPv4 tunnels), and potentially limited ICMP packets for connectivity testing such as echo-request or ICMP unreachable and time to live (TTL) expired messages for traceroute.

**Note:** Remember that ICMP is often used for simple denial-of-service (DoS) attacks and should only be permitted from external sources if necessary.

All RPs have a performance envelope in which they operate. Excessive traffic destined for the RP can overwhelm the router. This causes high CPU usage and ultimately results in packet and routing protocol drops that cause a denial of service. By filtering access to infrastructure routers from external sources, many of the external risks associated with a direct router attack are mitigated. Externally sourced attacks can no longer access infrastructure equipment. The attack is dropped on ingress interfaces into the autonomous system (AS).

The filtering techniques described in this document are intended to filter data destined for network infrastructure equipment. Do not confuse infrastructure filtering with generic filtering. The singular purpose of the infrastructure protection ACL is to restrict on a granular level what protocols and sources can access critical infrastructure equipment.

Network infrastructure equipment encompasses these areas:

- All router and switch management addresses, including loopback interfaces
- All internal link addresses: router-to-router links (point-to-point and multiple access)
- Internal servers or services that should not be accessed from external sources

In this document, all traffic not destined for the infrastructure is often referred to as transit traffic.

## Techniques

Infrastructure protection can be achieved through a variety of techniques:

- **Receive ACLs (rACLs)**

Cisco 12000 and 7500 platforms support rACLs that filter all traffic destined to the RP and do not affect transit traffic. Authorized traffic must be explicitly permitted and the rACL must be deployed on every router. Refer to GSR: Receive Access Control Lists for more information.

- **Hop-by-hop router ACLs**

Routers can also be protected by defining ACLs that permit only authorized traffic to the interfaces of the router, denying all others except for transit traffic, which must be explicitly permitted. This ACL is logically similar to an rACL but does affect transit traffic, and therefore can have a negative performance impact on the forwarding rate of a router.

- **Edge filtering via infrastructure ACLs**

ACLs can be applied to the edge of the network. In the case of a service provider (SP), this is the edge of the AS. This ACL explicitly filters traffic destined for infrastructure address space. Deployment of edge infrastructure ACLs requires that you clearly define your infrastructure space and the required/authorized protocols that access this space. The ACL is applied at ingress to your network on all externally facing connections, such as peering connections, customer connections, and so forth.

This document focuses on the development and deployment of edge infrastructure protection ACLs.

## ACL Examples

These IPv4 and IPv6 access lists provide simple yet realistic examples of typical entries required in a

protection ACL. These basic ACLs need to be customized with local site-specific configuration details. In dual IPv4 and IPv6 environments, both access-lists are deployed.

## IPv4 Example

```
!--- Anti-spoofing entries are shown here.

!--- Deny special-use address sources.
!--- Refer to RFC 3330 for additional special use addresses.

access-list 110 deny ip host 0.0.0.0 any
access-list 110 deny ip 127.0.0.0 0.255.255.255 any
access-list 110 deny ip 192.0.2.0 0.0.0.255 any
access-list 110 deny ip 224.0.0.0 31.255.255.255 any

!--- Filter RFC 1918 space.

access-list 110 deny ip 10.0.0.0 0.255.255.255 any
access-list 110 deny ip 172.16.0.0 0.15.255.255 any
access-list 110 deny ip 192.168.0.0 0.0.255.255 any

!--- Deny your space as source from entering your AS.
!--- Deploy only at the AS edge.

access-list 110 deny ip YOUR_CIDR_BLOCK any

!--- Permit BGP.

access-list 110 permit tcp host bgp_peer host router_ip eq bgp
access-list 110 permit tcp host bgp_peer eq bgp host router_ip

!--- Deny access to internal infrastructure addresses.

access-list 110 deny ip any INTERNAL_INFRASTRUCTURE_ADDRESSES

!--- Permit transit traffic.

access-list 110 permit ip any any
```

## IPv6 Example

The IPv6 access-list must be applied as an extended, named access-list.

```
!--- Configure the access-list.

ipv6 access-list iacl

!--- Deny your space as source from entering your AS.
!--- Deploy only at the AS edge.

deny ipv6 YOUR_CIDR_BLOCK_IPV6 any

!--- Permit multiprotocol BGP.

permit tcp host bgp_peer_ipv6 host router_ipv6 eq bgp
permit tcp host bgp_peer_ipv6 eq bgp host router_ipv6

!--- Deny access to internal infrastructure addresses.

deny ipv6 any INTERNAL_INFRASTRUCTURE_ADDRESSES_IPV6
```

```
!--- Permit transit traffic.
```

```
permit ipv6 any any
```

**Note:** The **log** keyword can be used to provide additional detail about source and destinations for a given protocol. Although this keyword provides valuable insight into the details of ACL hits, excessive hits to an ACL entry that uses the **log** keyword increases CPU utilization. The performance impact associated with logging varies by platform. Also, using the log keyword disables Cisco Express Forwarding (CEF) switching for packets that match the access-list statement. Those packets are fast switched instead.

## Develop a Protection ACL

In general, an infrastructure ACL is composed of four sections:

- Special-use address and anti-spoofing entries that deny illegitimate sources and packets with source addresses that belong within your AS from entering the AS from an external source

**Note:** RFC 3330 defines IPv4 special use addresses that might require filtering. RFC 1918 defines IPv4 reserved address space that is not a valid source address on the Internet. RFC 3513 defines the IPv6 addressing architecture. RFC 2827 provides ingress filtering guidelines.

- Explicitly permitted externally sourced traffic destined to infrastructure addresses
- **deny** statements for all other externally sourced traffic to infrastructure addresses
- **permit** statements for all other traffic for normal backbone traffic en route to noninfrastructure destinations

The final line in the infrastructure ACL explicitly permits transit traffic: **permit ip any any** for IPv4 and **permit ipv6 any any** for IPv6. This entry ensures that all IP protocols are permitted through the core and that customers can continue to run applications without issues.

The first step when you develop an infrastructure protection ACL is to understand the required protocols. Although every site has specific requirements, certain protocols are commonly deployed and must be understood. For instance, external BGP to external peers needs to be explicitly permitted. Any other protocols that require direct access to the infrastructure router need to be explicitly permitted as well. For example, if you terminate a GRE tunnel on a core infrastructure router, then protocol 47 (GRE) also needs to be explicitly permitted. Similarly, if you terminate an IPv6 over IPv4 tunnel on a core infrastructure router, then protocol 41 (IPv6 over IPv4) also needs to be explicitly permitted.

A classification ACL can be used to help identify the required protocols. The classification ACL is composed of **permit** statements for the various protocols that can be destined for an infrastructure router. Refer to the appendix on supported IP protocols in Cisco IOS® Software for a complete list. The use of the **show access-list command** to display a count of access control entry (ACE) hits identifies required protocols. Suspicious or surprising results must be investigated and understood before you create **permit** statements for unexpected protocols.

For example, this IPv4 ACL helps determine whether GRE, IPsec (ESP) and IPv6 tunneling (IP Protocol 41) need to be permitted.

```
access-list 101 permit GRE any infrastructure_ips
access-list 101 permit ESP any infrastructure_ips
access-list 101 permit 41 any infrastructure_ips
access-list 101 permit ip any infrastructure_ips log
```

```
!--- The log keyword provides more details
!--- about other protocols that are not explicitly permitted.
```

```
access-list 101 permit ip any any

interface <int>
 ip access-group 101 in
```

This IPv6 ACL can be used to determine if GRE and IPsec (ESP) need to be permitted.

```
ipv6 access-list determine_protocols
 permit GRE any infrastructure_ips_ipv6
 permit ESP any infrastructure_ips_ipv6
 permit ipv6 any infrastructure_ips_ipv6 log

!--- The log keyword provides more details
!--- about other protocols that are not explicitly permitted.

 permit ipv6 any any

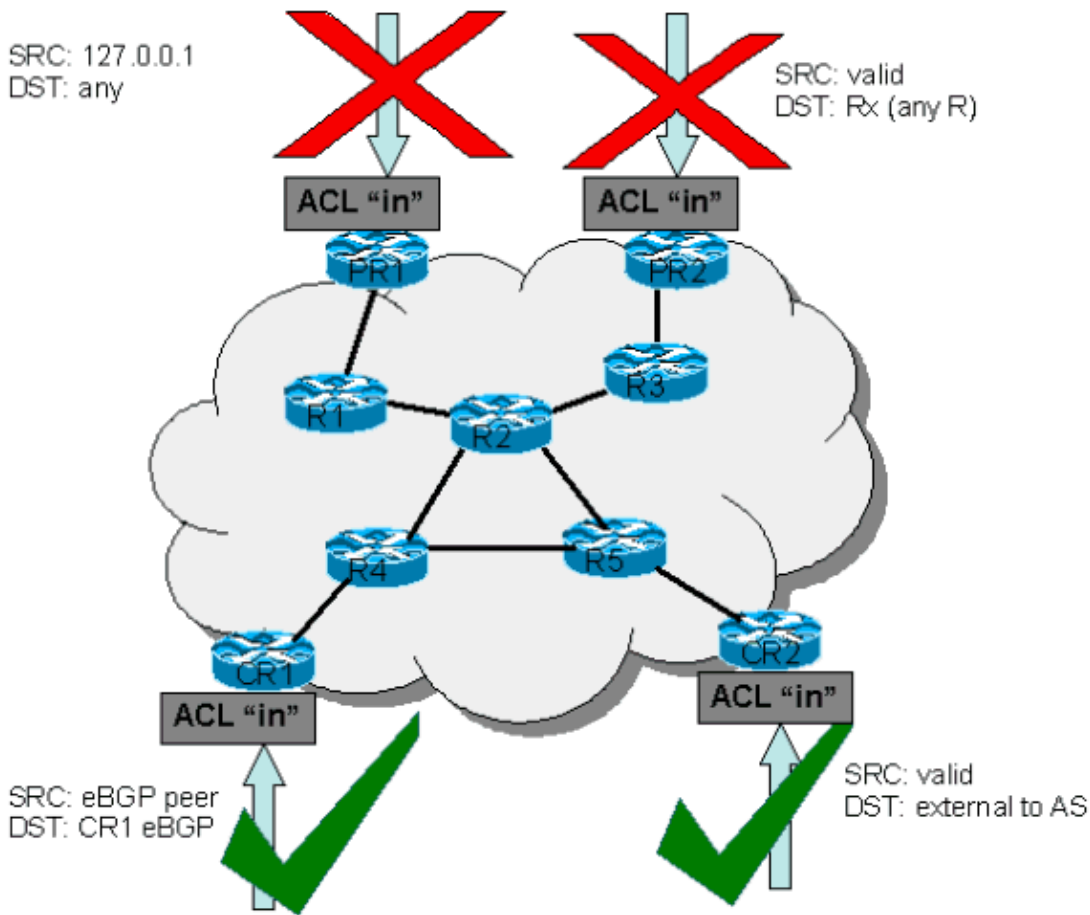
interface <int>
 ipv6 traffic-filter determine_protocols in
```

In addition to required protocols, infrastructure address space needs to be identified since this is the space the ACL protects. Infrastructure address space includes any addresses that are used for the internal network and are rarely accessed by external sources such as router interfaces, point-to-point link addressing, and critical infrastructure services. Since these addresses are used for the destination portion of the infrastructure ACL, summarization is critical. Wherever possible, these addresses must be grouped into classless interdomain routing (CIDR) blocks.

With the use of the protocols and addresses identified, the infrastructure ACL can be built to permit the protocols and protect the addresses. In addition to direct protection, the ACL also provides a first line of defense against certain types of invalid traffic on the Internet.

- RFC 1918 space must be denied.
- Packets with a source address that fall under special-use address space, as defined in RFC 3330, must be denied.
- Anti-spoof filters must be applied. (Your address space must never be the source of packets from outside your AS.)

This newly constructed ACL must be applied inbound on all ingress interfaces. See the sections on deployment guidelines and deployment examples for more details.



## ACLs and Fragmented Packets

ACLs have a **fragments** keyword that enables specialized fragmented packet-handling behavior. Without this **fragments** keyword, noninitial fragments that match the Layer 3 statements (irrespective of the Layer 4 information) in an ACL are affected by the permit or deny statement of the matched entry. However, by adding the **fragments** keyword, you can force ACLs to either deny or permit noninitial fragments with more granularity. This behavior is the same for both IPv4 and IPv6 access-lists, with the exception that, while IPv4 ACLs allow the use of the fragments keyword within Layer 3 and Layer 4 statements, IPv6 ACLs only allow the use of the fragments keyword within Layer 3 statements.

Filtering fragments adds an additional layer of protection against a Denial of Service (DoS) attack that uses noninitial fragments (that is, FO > 0). Using a **deny** statement for noninitial fragments at the beginning of the ACL denies all noninitial fragments from accessing the router. Under rare circumstances, a valid session might require fragmentation, and therefore be filtered if a **deny fragment** statement exists in the ACL.

For example, consider this partial IPv4ACL:

```
access-list 110 deny tcp any infrastructure_IP fragments
access-list 110 deny udp any infrastructure_IP fragments
access-list 110 deny icmp any infrastructure_IP fragments
<rest of ACL>
```

The addition of these entries to the beginning of an ACL denies any noninitial fragment access to the core routers, while nonfragmented packets or initial fragments pass to the next lines of the ACL unaffected by the **deny fragment** statements. The preceding ACL command also facilitates classification of the attack since each protocol Universal Datagram Protocol (UDP), TCP, and ICMP increments separate counters in the ACL.

This is a comparable example for IPv6:

```
ipv6 access-list iacl
deny ipv6 any infrastructure_IP fragments
```

The addition of this entry to the beginning of an IPv6 ACL denies any noninitial fragment access to the core routers. As previously noted, IPv6 access-lists only allow the use of the fragments keyword within Layer 3 statements.

Since many attacks rely on flooding core routers with fragmented packets, filtering incoming fragments to the core infrastructure provides an added measure of protection and helps ensure that an attack cannot inject fragments by simply matching Layer 3 rules in the infrastructure ACL.

Refer to Access Control Lists and IP Fragments for a detailed discussion of the options.

## Risk Assessment

Consider these two areas of key risk when you deploy infrastructure protection ACLs:

- Ensure that the appropriate **permit/deny** statements are in place. For the ACL to be effective, all required protocols must be permitted and the correct address space must be protected by the **deny** statements.
- ACL performance varies from platform to platform. Review the performance characteristics of your hardware before you deploy ACLs.

As always, it is recommended that you test this design in the lab prior to deployment.

## Appendices

### Supported IP Protocols in Cisco IOS Software

These IP protocols are supported by Cisco IOS Software:

- 1 ICMP
- 2 IGMP
- 3 GGP
- 4 IP in IP encapsulation
- 6 TCP
- 8 EGP
- 9 IGRP
- 17 UDP
- 20 HMP
- 27 RDP
- 41 IPv6 in IPv4 tunneling
- 46 RSVP
- 47 GRE
- 50 ESP
- 51 AH
- 53 SWIPE
- 54 NARP
- 55 IP mobility
- 63 any local network

- 77 Sun ND
- 80 ISO IP
- 88 EIGRP
- 89 OSPF
- 90 Sprite RPC
- 91 LARP
- 94 KA9Q/NOS compatible IP over IP
- 103 PIM
- 108 IP compression
- 112 VRRP
- 113 PGM
- 115 L2TP
- 120 UTI
- 132 SCTP

## Deployment Guidelines

Cisco recommends conservative deployment practices. In order to successfully deploy infrastructure ACLs, required protocols must be well understood, and address space must be clearly identified and defined. These guidelines describe a very conservative method for deploying protection ACLs using an iterative approach.

### 1. Identify protocols used in the network with a classification ACL.

Deploy an ACL that permits all the known protocols that access infrastructure devices. This discovery ACL has a source address of **any** and a destination that encompasses infrastructure IP space. Logging can be used to develop a list of source addresses that match the protocol **permit** statements. A last line permitting **ip any any** (IPv4) or **ipv6 any any** (IPv6) is required to permit traffic flow.

The objective is to determine what protocols the specific network uses. Logging is used for analysis to determine what else might be communicating with the router.

**Note:** Although the **log** keyword provides valuable insight into the details of ACL hits, excessive hits to an ACL entry that uses this keyword might result in an overwhelming number of log entries and possibly high router CPU usage. Also, using the **log** keyword disables Cisco Express Forwarding (CEF) switching for packets that match the access-list statement. Those packets are fast switched instead. Use the **log** keyword for short periods of time and only when needed to help classify traffic.

### 2. Review identified packets and begin to filter access to the route processor RP.

Once the packets filtered by the ACL in step 1 have been identified and reviewed, deploy an ACL with a **permit any source** to infrastructure addresses for the allowed protocols. Just as in step 1, the **log** keyword can provide more information about the packets that match the **permit** entries. Using **deny any** at the end can help identify any unexpected packets destined to the routers. The last line of this ACL must be a **permit ip any any** (IPv4) or **permit ipv6 any any** (IPv6) statement to permit the flow of transit traffic. This ACL does provide basic protection and does allow network engineers to ensure that all required traffic is permitted.

### 3. Restrict source addresses.

Once you have a clear understanding of the protocols that must be permitted, further filtering can be performed to allow only authorized sources for those protocols. For example, you can explicitly permit external BGP neighbors or specific GRE peer addresses.

This step narrows the risk without breaking any services and allows you to apply granular control to sources that access your infrastructure equipment.

### 4. Limit the destination addresses on the ACL. (*optional*)



```

!--- Phase 2   Explicit Permit
!--- Permit only applications/protocols whose destination
!--- address is part of the infrastructure IP block.
!--- The source of the traffic should be known and authorized.

!

!--- Note: This template must be tuned to the network s
!--- specific source address environment. Variables in
!--- the template need to be changed.

!--- Permit external BGP.

access-list 110 permit tcp host 169.254.254.1  host 169.223.252.1  eq bgp
access-list 110 permit tcp host 169.254.254.1  eq bgp host 169.223.252.1
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 3   Explicit Deny to Protect Infrastructure

access-list 110 deny ip any 169.223.252.0 0.0.3.255
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 4   Explicit Permit for Transit Traffic

access-list 110 permit ip any any

```

## IPv6 Example

This IPv6 example shows an Infrastructure ACL protecting a router based on this addressing:

- The overall prefix block allocated to the ISP is 2001:0DB8::/32.
- The IPv6 prefix block used by the ISP for network infrastructure addresses is 2001:0DB8:C18::/48.
- There is a BGP peering router with a source IPv6 address of 2001:0DB8:C18:2:1::1 that peers to destination IPv6 address of 2001:0DB8:C19:2:1::F.

The infrastructure protection ACL displayed is developed based on the preceding information. The ACL permits external multiprotocol BGP peering to the external peer, provides anti-spoof filters, and protects the infrastructure from all external access.

```

no ipv6 access-list iacl
ipv6 access-list iacl
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 1   Anti-spoofing and Fragmentation Denies
!--- These ACEs deny fragments and spoofs of
!--- internal space as an external source.
!--- Deny fragments to the infrastructure block.

deny ipv6 any 2001:0DB8:C18::/48 fragments

!--- Deny our internal space as an external source.
!--- This is only deployed at the AS edge.

```

```

deny ipv6 2001:0DB8::/32 any
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 2   Explicit Permit
!--- Permit only applications/protocols whose destination
!--- address is part of the infrastructure IP block.
!--- The source of the traffic should be known and authorized.

!--- Note: This template must be tuned to the
!--- specific source address environment of the network. Variables in
!--- the template need to be changed.

!--- Permit multiprotocol BGP.

permit tcp host 2001:0DB8:C19:2:1::F host 2001:0DB8:C18:2:1::1 eq bgp
permit tcp host 2001:0DB8:C19:2:1::F eq bgp host 2001:0DB8:C18:2:1::1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 3   Explicit Deny to Protect Infrastructure

deny ipv6 any 2001:0DB8:C18::/48
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 4   Explicit Permit for Transit Traffic

permit ipv6 any any

```

## NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for Security
Security: Intrusion Detection [Systems]
Security: AAA
Security: General
Security: Firewalling

## Related Information

- [Access Lists Support Page](#)
- [Requests for Comments \(RFCs\)](#)
- [Technical Support & Documentation – Cisco Systems](#)

