

Table of Contents

<u>Protecting Your Core: Infrastructure Protection Access Control Lists</u>	1
<u>Introduction</u>	1
<u>Infrastructure Protection</u>	1
<u>Background</u>	1
<u>Techniques</u>	2
<u>ACL Example</u>	2
<u>Developing a Protection ACL</u>	3
<u>ACLs and Fragmented Packets</u>	5
<u>Risk Assessment</u>	6
<u>Appendices</u>	6
<u>Supported IP Protocols in Cisco IOS Software</u>	6
<u>Deployment Guidelines</u>	7
<u>Deployment Example</u>	7
<u>Related Information</u>	9

Protecting Your Core: Infrastructure Protection Access Control Lists

Introduction

Infrastructure Protection

- Background

- Techniques

ACL Example

Developing a Protection ACL

ACLs and Fragmented Packets

Risk Assessment

Appendices

- Supported IP Protocols in Cisco IOS Software

- Deployment Guidelines

- Deployment Example

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 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 and 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 (such as Simple Network Management Protocol [SNMP]). In addition, protocols such as Internet Control Message Protocol (ICMP) and IP options might 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 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. Please bear in mind 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, causing high CPU usage and ultimately resulting 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 direct router attack are mitigated. Externally sourced attacks can no longer access infrastructure equipment; the attack is simply 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 infrastructure protection ACL has a singular purpose: to restrict on a granular level what protocols and sources can access critical infrastructure equipment.

Network infrastructure equipment encompasses the following.

- 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. For more information, refer to GSR: Receive Access Control Lists

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

Routers can also be protected by defining ACLs that permit only authorized traffic to the router's interfaces, 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 a router's forwarding rate.

- **Edge filtering via infrastructure ACLs**

ACLs can be applied to the edge of the network; in the case of a service provider (SP), this would be the edge of the AS. This ACL will explicitly filter 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, etc.).

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

ACL Example

The following access list provides a simple yet realistic example of typical entries required in a protection ACL. This basic ACL needs to be customized with local site-specific configuration details.

!--- 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

!--- 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 your space as source.

access-list 110 deny ip YOUR_CIDR_BLOCK any

!--- 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

```

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 will increase CPU utilization. The performance impact associated with logging will vary with by platform.

Developing 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 special use addresses that might require filtering. RFC 1918 defines reserved address space that is not a valid source address on the Internet. 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 will explicitly permit transit traffic: **permit ip any any**. 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 developing an infrastructure protection ACL is to understand the required protocols.

Although every site has specific requirements, certain protocols are commonly deployed and will be well understood. For instance, external BGP to external peers needs to be explicitly permitted. Any other protocols that require direct access to the infrastructure router will 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.

To help identify the required protocols, a classification ACL should be used. The classification ACL is composed of **permit** statements for the various protocols that could be destined for an infrastructure router. (Refer to the appendix on supported IP protocols in Cisco IOS® Software for a complete list.) Using the **show access-list command** to display a count of access control entry (ACE) hits will identify required protocols. Suspicious or surprising results should be investigated and well understood prior to creating **permit** statements for unexpected protocols.

For example, the following ACL would help determine whether GRE, IPsec (ESP) and IPv6 tunnelling 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 an 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

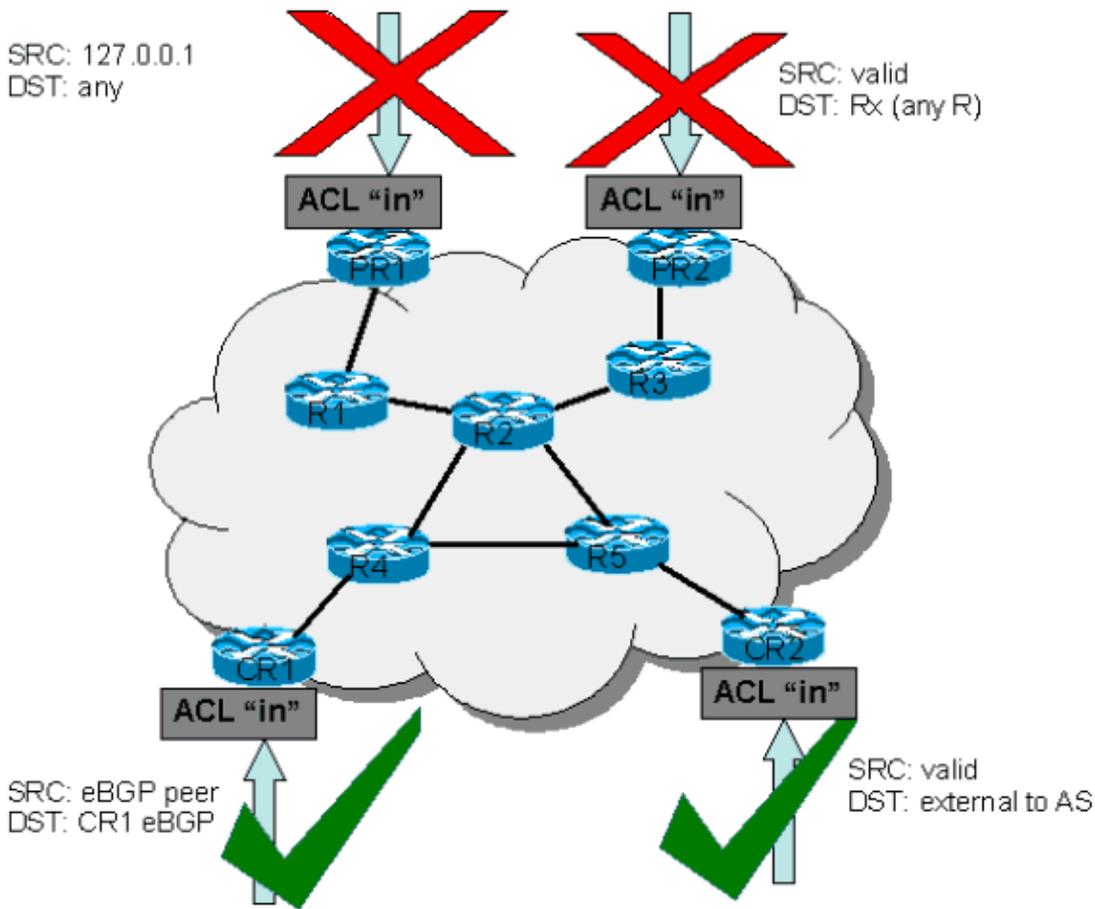
interface <int>
    ip access-group 101 in
```

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

Using 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 should also provide a first line of defence against certain types of invalid traffic on the Internet.

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

This newly constructed ACL should be applied inbound on all ingress interfaces. For more details, refer to the sections on deployment guidelines and deployment example.



ACLs and Fragmented Packets

ACLs have a **fragments** keyword that enables specialized fragmented packet-handling behavior. In general, noninitial fragments that match the L3 statements (irrespective of the L4 information) in an ACL are affected by the **permit** or **deny** statement of the matched entry. Note that the use of the **fragments** keyword can force ACLs to either deny or permit noninitial fragments with more granularity.

Filtering fragments adds an additional layer of protection against a DoS attack that uses only noninitial fragments (such as 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 the partial ACL shown below.

```
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>
```

Adding 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 above ACL snippet also facilitates classification of the attack since each protocol Universal Datagram Protocol (UDP), TCP, and ICMP increments separate counters in the ACL.

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.

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

Risk Assessment

When deploying infrastructure protection ACLs, remember to consider two key areas of risk.

- 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. Before deploying ACLs, review the performance characteristics of your hardware.

As always, Cisco recommends that you test this design in the lab prior to deployment.

Appendices

Supported IP Protocols in Cisco IOS Software

The following 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 IP 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. To successfully deploy infrastructure ACLs, required protocols must be well understood, and address space must be clearly identified and defined. The following guidelines describe a very conservative method for deploying protection ACLs using 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 should have 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** is required to permit traffic flow.

The objective is to determine what protocols the specific network uses. Logging should be 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. 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 GRP.

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 should be a **permit ip any any** statement to permit the flow of transit traffic. This ACL will provide basic protection and will 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)*

Some Internet service providers (ISP) may choose to only allow specific protocols to use specific destination addresses on the router. This final phase is meant to limit the range of destination addresses that will accept traffic for a protocol.

Deployment Example

The example below shows a receive ACL protecting a router based on the following addressing.

- The ISP s address block is 169.223.0.0/16.
- The ISP s infrastructure block is 169.223.252.0/22.
- The loopback for the router is 169.223.253.1/32.
- The router is a peering router and peers with 10.1.1.1 (to address 10.1.1.2).

The infrastructure protection ACL shown below was developed based on this information. The ACL permits external BGP peering to the external peer, provides anti-spoof filters, and protects the infrastructure from all external access.

```

!
no access-list 110
!
! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 1   Anti-spoofing Denies
!--- These ACEs deny fragments, RFC 1918 space,
!--- invalid source addresses, and spoofs of
!--- internal space (space as an external source).

!

!--- Deny fragments.

access-list 110 deny tcp any 169.223.252.0 0.0.252.255 fragments
access-list 110 deny udp any 169.223.252.0 0.0.252.255 fragments
access-list 110 deny icmp any 169.223.252.0 0.0.252.255 fragments

!--- Deny special-use address sources.
!--- See 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
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- 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 10.1.1.1 host 10.1.1.2 eq bgp
access-list 110 permit tcp host 10.1.1.1 eq bgp host 10.1.1.2
!

```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 3  Explicit Deny to Protect Infrastructure

access-list 110 deny ip any 169.223.252.0 0.0.252.255
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!--- Phase 4  Explicit Permit for Transit Traffic

access-list 110 permit ip any any
```

Related Information

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

All contents are Copyright © 1992–2003 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.