IP Access Lists

IP access control lists

An IP Access Control List (ACL) is a network traffic filtering mechanism that

- consists of ordered permit and deny statements evaluating IP addresses, protocols, and directions,
- controls packets by allowing or blocking their movement based on matching criteria,
- and is applied to interfaces, affecting traffic entering or leaving a system, but not traffic originating from the system itself.

ACLs are sequential lists that contain permit or deny statements for IP addresses and upper-layer protocols. These lists control packet flow by matching access list parameters to the information in the packet header. An access list becomes effective only when it is created and applied to an interface.

Packets can be filtered when they arrive at the device (ingress) or leave the device (egress). However, access lists cannot control traffic that originates from the system.

ACL processing paths:

There are two paths for interface packet filtering:

- Hardware programming path (fast path): Uses Ternary Content Addressable Memory (TCAM) via the packet filter Execution Agent for rapid ACL processing.
- Software programming path (slow path): Requires additional configuration through Interface Manager and NetIO, used for management interfaces.

Statistics for ACLs are collected separately for fast path and slow path packets. ACL information is stored globally on the route processor.

Examples of supported features in Cisco NCS 1004:

- Ingress ACL is supported for both IPv4 and IPv6.
- Management interface uses the slow packet path for ACL processing.
- Egress ACL: Self-originated packets are not controlled by ACLs, as these are already managed by the user. ACLs only filter forwarded packets/traffic, for both IPv4 and IPv6.

Best practices and requirements for configuring IP access control lists

These best practices, requirements, and guidelines should be followed when configuring IP ACLs.

- Requirement: Always include at least one permit statement in an access list; otherwise, all packets are denied.
- Best practice: Pay careful attention to the order of permit and deny statements. The order is critical because different orders may lead to different packet outcomes.
- Requirement: Only one access list is allowed for each interface, protocol, and direction.
- Best practice: Apply access lists only to management interfaces. ACLs do not filter traffic on other interfaces or controllers.
- Best practice: Always create the access list before attempting to apply it to an interface using the access-group command.
- Caution: To remove an access list, first remove its reference from the access group, and then remove the access list itself.
- Note: IP access lists cannot filter packets originating from the device itself; they filter only packets arriving at or leaving through an interface.

When you follow these best practices, IP access lists function as intended and protect network resources, preventing unintended packet loss and security breaches.

Configure an IP ACL

Use this procedure whenever you need to restrict or allow specific traffic types entering or leaving a network interface by specifying detailed access list rules.

Procedure

Step 1 Use the command **interface** *interface-type Rack/Slot/Instance/Port* to configure the interface in global configuration mode.

Example:

```
RP/0/RP0/CPU0:ios#configure
Fri Oct 20 05:25:58.785 UTC
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1t
```

Step 2 Use the command **ipv4 address** or **ipv6 address** in the interface configuration mode to configure IPv4 or IPv6 address for the interface.

Example:

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#ipv6 address 1000::1/64
```

Step 3 Use the **ipv4 | ipv6 access-group access-list-name {ingress | egress}** command, to configure the ACL at the IPv4 or IPv6 interface in the interface configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 access-group IPV4_ICMP_DENY ingress
RP/0/RP0/CPU0:ios(config-if)#ipv4 access-group IPV4_ROUTER_FWD_TELNET_TRAFFIC_DENY egress
RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group IPV6_SSH_DENY ingress
RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group IPV6_ROUTER_FWD_TELNET_TRAFFIC_DENY egress
```

Step 4 Commit the configuration.

Example:

```
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
RP/0/RP0/CPU0:ios(config-ipv6-acl)# commit
```

The IP access list is successfully applied to the selected interface. Traffic entering or leaving the interface is now filtered according to the rules you have defined for IPv4 and IPv6, helping enforce your network security policy.

Verify access control lists

Monitoring the number of packets processed by access control lists (ACLs) helps maintain network security and troubleshoot traffic flow. Use this task to review ACL statistics for both IPv4 and IPv6 to ensure filters are working as expected.

Procedure

Use the **show access-lists ipv4** or **show access-lists ipv6** command to verify the IPv4 or IPv6 ACLs.

Example:

IPv4

```
RP/0/RP0/CPU0:ios#show access-lists ipv4
Wed Jan 17 09:52:12.448 IST
ipv4 access-list IPV4_ICMP_DENY
10 deny icmp any any (8 matches)
20 permit ipv4 any any (106 matches)
ipv4 access-list IPV4_ROUTER_FWD_TELNET_TRAFFIC_DENY
10 deny tcp any any eq telnet (3 matches)
20 permit ipv4 any any (6 matches)
```

Example:

IPv6

```
RP/0/RP0/CPU0:ios#show access-lists ipv6
Wed Jan 17 09:52:14.591 IST
ipv6 access-list IPV6_ROUTER_FWD_TELNET_TRAFFIC_DENY
10 deny tcp any any eq telnet (3 matches)
20 permit ipv6 any any (5 matches)
ipv6 access-list IPV6_SSH_DENY
10 deny tcp any any eq ssh (9 matches)
20 permit ipv6 any any (100 matches)
```