

Block CPU-Bound Traffic to Loopback via ACL

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Q. Can you block CPU-bound traffic \(such as ICMP\) destined towards a Loopback interface via an Access Control List \(ACL\)?](#)

[A. No. ACLs applied to Loopback interfaces do not block traffic that is destined to the control plane of the router, that is, punted traffic.](#)

Introduction

This document describes a limitation in blocking CPU-bound traffic via an ACL applied on a Loopback interface.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Software-Defined Wide Area Network (SD-WAN)

Components Used

The information in this document is based on these software and hardware versions:

- C8000V version 17.12.2
- vManage version 20.12.2

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.

Q. Can you block CPU-bound traffic (such as ICMP) destined towards a Loopback interface via an Access Control List (ACL) ?



Note: This answer is applicable to Controller, Autonomous, and SD-Routing mode Cisco IOS® routers. For Controller mode devices, this answer applies to Explicit ACLs in policy or Cisco IOS config.

A. No. ACLs applied to Loopback interfaces do not block traffic that is destined to the control plane of the router, that is, punted traffic.

This is because the router, realizing that any traffic destined towards the Loopback IP is destined for the control plane, programs the hardware to send the traffic directly to the CPU and bypass the Loopback interface all together for efficiency. This means that anything that gets applied on ingress of the Loopback interface (for instance, ACLs) are not triggered since the traffic never technically ingresses the Loopback interface. You can verify the hardware programming via a Cisco Express Forwarding® (CEF) command.

```
Edge#show ip route 10.0.0.1
Routing entry for 10.0.0.1/32
  Known via "connected", distance 0, metric 0 (connected)
  Routing Descriptor Blocks:
```

* directly connected, via Loopback1
Route metric is 0, traffic share count is 1

Edge#show ip cef exact-route 172.16.0.1 10.0.0.1 protocol 1
172.16.0.1 -> 10.0.0.1 =>receive <<< no mention of Loopback1

If we take a FIA Trace on a ping packet, we see that the traffic is sent to the CPU and the ACL is not even hit.

Edge#show platform packet-trace packet 0 decode
Packet: 0 CBUG ID: 570
Summary
Input : GigabitEthernet1
Output : internal0/0/rp:0
State : PUNT 11 (For-us data)
Timestamp
Start : 1042490936823469 ns (11/26/2024 16:41:12.259675 UTC)
Stop : 1042490936851807 ns (11/26/2024 16:41:12.259703 UTC)
Path Trace
Feature: IPV4(Input)
Input : GigabitEthernet1
Output : <unknown>
Source : 172.16.0.1
Destination : 10.0.0.1
Protocol : 1 (ICMP)
<... output omitted ...>
Feature: SDWAN Implicit ACL
Action : ALLOW
Reason : SDWAN_SERV_ALL
<... output omitted ...>
Feature: IPV4_INPUT_LOOKUP_PROCESS_EXT
Entry : Input - 0x814f8e80
Input : GigabitEthernet1
Output : internal0/0/rp:0
Lapsed time : 2135 ns
<... output omitted ...>
Feature: INTERNAL_TRANSMIT_PKT_EXT
Entry : Output - 0x814cb454
Input : GigabitEthernet1
Output : internal0/0/rp:0
Lapsed time : 5339 ns

IOSd Path Flow: Packet: 0 CBUG ID: 570
Feature: INFRA
Pkt Direction: IN
Packet Rcvd From DATAPLANE

Feature: IP
Pkt Direction: IN
Packet Enqueued in IP layer
Source : 172.16.0.1
Destination : 10.0.0.1
Interface : GigabitEthernet1

Feature: IP
Pkt Direction: IN
FORWARDED To transport layer
Source : 172.16.0.1

```
Destination   : 10.0.0.1
Interface     : GigabitEthernet1
```

```
Edge#show platform packet-trace packet 0 decode | in ACL <<<< ACL feature never hit
Feature: SDWAN Implicit ACL
Feature: IPV4_SDWAN_IMPLICIT_ACL_EXT
```

```
Edge#show platform packet-trace packet 0 decode | in Lo <<<< Loopback1 never mentioned
Edge#
```

In order to block CPU-bound traffic, you need to apply the ACL to the interface that the packet first ingresses, for example, the physical interface or port channel . Here, we can see the result of applying the ACL on the physical interface.

```
Edge1#show platform packet-trace packet 0
Packet: 0          CBUG ID: 24
Summary
  Input      : GigabitEthernet1
  Output     : GigabitEthernet1
  State      : DROP 8    (Ipv4Ac1)
Timestamp
  Start     : 5149395094183 ns (11/27/2024 19:48:55.202545 UTC)
  Stop      : 5149395114474 ns (11/27/2024 19:48:55.202565 UTC)
Path Trace
Feature: IPV4(Input)
  Input      : GigabitEthernet1
  Output     : <unknown>
  Source     : 172.16.0.1
  Destination : 10.0.0.1
  Protocol   : 1 (ICMP)
<... output omitted ...>
Feature: IPV4_INPUT_ACL <<<<
  Entry      : Input - 0x814cc220
  Input      : GigabitEthernet1
  Output     : <unknown>
  Lapsed time : 15500 ns
```