Use a Static Route to the Null0 Interface for Loop Prevention

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

Command Syntax

Example

- Packet Flow

Related Information

Introduction

This document describes how a Static Route to the Null interface can prevent routing loops.

Prerequisites

Requirements

There are no specific prerequisites for this document.

Components Used

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

- Cisco IOS®Software Release 12.3.

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.

Background Information

The Null interface is typically used for preventing routing loops. Enhanced Interior Gateway Routing Protocol (EIGRP), for instance, always creates a route to the Null0 interface when it summarizes a group of routes. Whenever a routing protocol summarizes, it means that the router can receive traffic for any IP address within that summary.

Because not all IP addresses are always in use, there is a risk of looping packets in case default routes are used on the router which receives the traffic for the summary.

Command Syntax

A static route to Null0 is a normal static route, except that it points to the Null0 interface, which is a virtual Cisco IOS interface.

Refer to the ip route section of Chapter: IP Routing Protocol-Independent Commands A through R for more information about the **ip route** command.

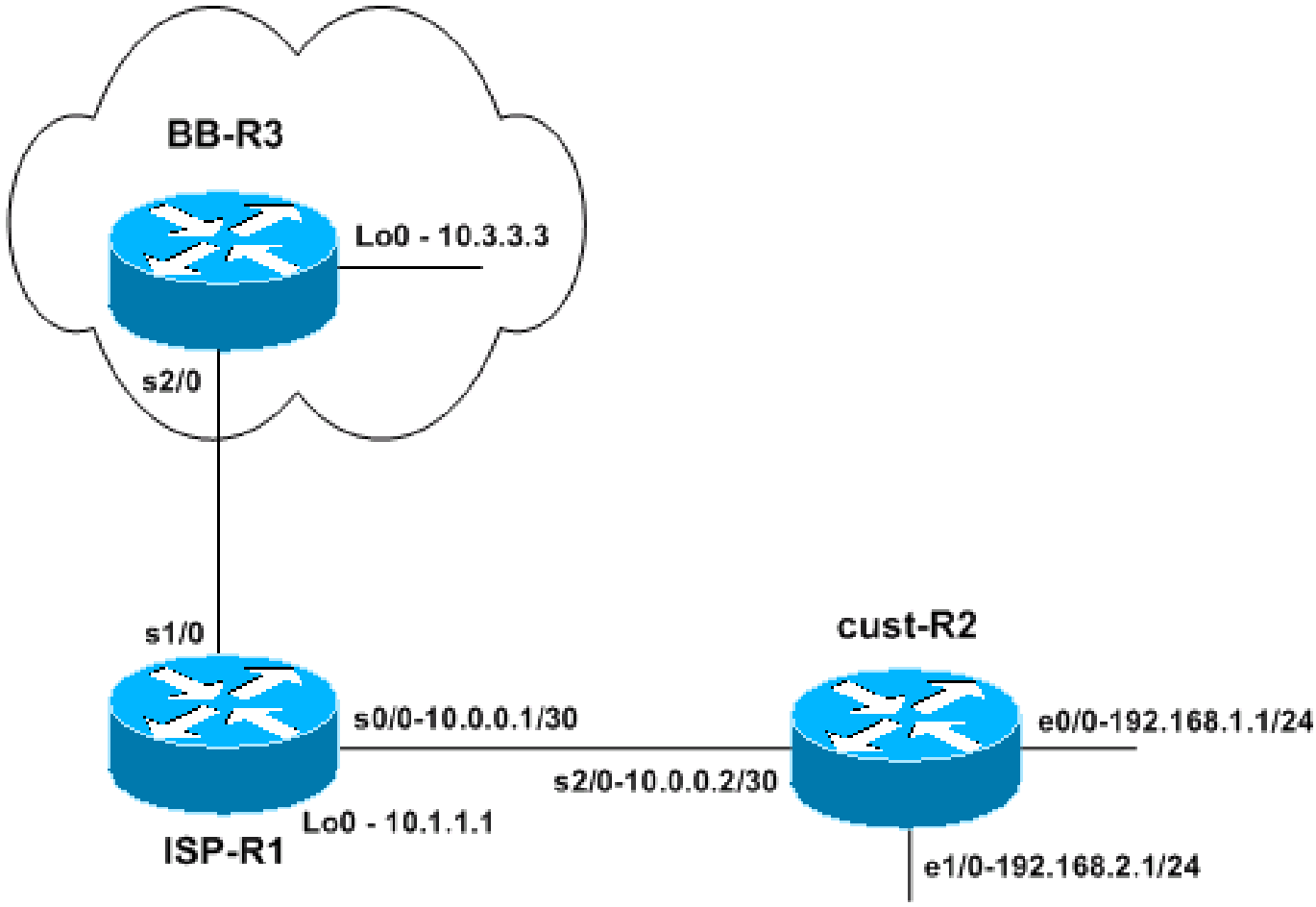
The next section presents an example of how to use the **ip route** command to create a static route to Null0.

Example

A common scenario where you can need to add a static route to Null0 is that of an access server which has many clients dialing in. This scenario causes host routes to be installed in the access server routing table.

To ensure reachability to these clients, while not flooding the entire network with host routes, other routers in the network typically have a summary route which points to the access server. In this type of configuration, the access server must have that same summary route that points to the access server Null0 interface. If not, routing loops can occur when outside hosts attempt to reach IP addresses not currently assigned to a dialed in client but are part of the summary route. This is because the access server would bounce the packets back over the access server default route into the core network, because the access server lacks a specific host route for the destination.

Consider this example:



Network Topology

A small ISP (ISP-R1) gives one of the users a network block of 192.168.0.0/16. In this example, the user divided 192.168.0.0/16 in /24 networks and only uses 192.168.1.0/24 and 192.168.2.0/24 at the moment. On

router ISP-R1, the ISP configures a static route for 192.168.0.0/16 toward the user router (cust-R2). The ISP then connects to a backbone ISP, represented by router BB-R3. Router BB-R3 sends a default route to ISP-R1 and receives the network 192.168.0.0/16 via BGP from ISP-R1.

Reachability is now guaranteed from the Internet (backbone ISP router BB-R3) to the user router cust-R2 because cust-R2 has a default route configured to point to ISP-R1.

However, if packets are destined to network blocks which are not in use out of the 192.168.0.0/16 range, then the cust-R2 router uses the default route to ISP-R1 to forward those packets.

The packs then loop between ISP-R1 and cust-R2 until the TTL expires. This can have a huge impact on the router CPU and link utilization.

An example of where this traffic to unused IP addresses can come from could be denial of service attacks, scanning of IP blocks to find vulnerable hosts, and so on.

Relevant configurations:

cust-R2

```
version 12.3
!
hostname cust-R2
!
ip subnet-zero
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.2.1 255.255.255.0
!
interface Serial2/0
 ip address 10.0.0.2 255.255.255.252

!--- This interface leads to ISP-R1.

!
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.0.1

!--- Default route going to ISP-R1.

!
end
```

ISP-R1

```
version 12.3
!
hostname ISP-R1
!
ip subnet-zero
```

```
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface Serial0/0
 ip address 10.0.0.1 255.255.255.252

!--- Interface to cust-R2.

!
interface Serial1/0
 ip unnumbered Loopback0

!--- Interface going to BB-R3.

!
router bgp 65501
 no synchronization
 network 192.168.0.0 mask 255.255.0.0

!--- ISP-R1 injects 192.168.0.0/16 into BGP to
!--- advertise it to BB-R3.

 neighbor 10.3.3.3 remote-as 65503
 neighbor 10.3.3.3 ebgp-multihop 255
 no auto-summary
!
ip classless
ip route 10.3.3.3 255.255.255.255 Serial1/0
ip route 192.168.0.0 255.255.0.0 Serial0/0

!--- The first route is necessary for the eBGP
!--- session to BB-R3 to come up.
!--- The route to 192.168.0.0/16 points towards cust-R2.

!
!
end
```

BB-R3

```
version 12.3
!
hostname BB-R3
!
ip subnet-zero
!
!
interface Loopback0
 ip address 10.3.3.3 255.255.255.255
!
interface Serial2/0
 ip unnumbered Loopback0

!--- This interface goes to ISP-R1.

!
router bgp 65503
 no synchronization
 bgp log-neighbor-changes
```

```
 neighbor 10.1.1.1 remote-as 65501
 neighbor 10.1.1.1 ebgp-multihop 255
 neighbor 10.1.1.1 default-originate

!--- BB-R3 injects a default route into BGP and
!--- sends it to ISP-R1.

 no auto-summary
!
ip classless
ip route 10.1.1.1 255.255.255.255 Serial2/0

!--- This route points to ISP-R1 and is
!--- used to establish the eBGP peering.

!
end
```

Packet Flow

Note: Some **debug** commands were enabled on the routers to better illustrate the packet flow, notably **debug ip packet** and **debug ip icmp**. Do not enable these commands in a production environment unless you fully understand the consequences.

```
<#root>

BB-R3#

ping ip 192.168.20.1 repeat 1


Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:

*Oct  6 09:36:45.355: IP: tableid=0, s=10.3.3.3 (local), d=192.168.20.1 (Serial2/0), routed via FI
*Oct  6 09:36:45.355: IP: s=10.3.3.3 (local), d=192.168.20.1 (Serial2/0), len 100, sending.
Success rate is 0 percent (0/1)
BB-R3#
*Oct  6 09:36:50.943: ICMP: time exceeded rcvd from 10.0.0.1
```

BB-R3 sends a single ICMP request to an IP address within the 192.168.0.0/16 block which is not in use on cust-R2.

BB-R3 receives an ICMP time exceeded back from ISP-R1.

On ISP-R1:

```
18:50:22: IP: tableid=0, s=10.3.3.3 (Serial1/0), d=192.168.20.1 (Serial0/0), routed via RIB
18:50:22: IP: s=10.3.3.3 (Serial1/0), d=192.168.20.1 (Serial0/0), g=192.168.20.1, len 100, forward
18:50:22: IP: tableid=0, s=10.3.3.3 (Serial0/0), d=192.168.20.1 (Serial0/0), routed via RIB
18:50:22: IP: s=10.3.3.3 (Serial0/0), d=192.168.20.1 (Serial0/0), g=192.168.20.1, len 100, forward
18:50:22: IP: tableid=0, s=10.3.3.3 (Serial0/0), d=192.168.20.1 (Serial0/0), routed via RIB
```

```
18:50:22: IP: s=10.3.3.3 (Serial0/0), d=192.168.20.1 (Serial0/0), g=192.168.20.1, len 100, forward
18:50:22: IP: tableid=0, s=10.3.3.3 (Serial0/0), d=192.168.20.1 (Serial0/0), routed via RIB
```

The initial packet is received on serial1/0 on ISP-R1 from BB-R3 and is forwarded to cust-R2 on serial0/0 as expected.

The same packet arrives back at ISP-R1 on serial0/0 and is sent immediately out the same interface, to cust-R2, because of this route:

```
<#root>

ISP-R1#

show ip route 192.168.20.1

Routing entry for 192.168.0.0/16, supernet
  Known via "static", distance 1, metric 0 (connected)
  Advertised by bgp 65501
  Routing Descriptor Blocks:
  * directly connected, via Serial0/0
      Route metric is 0, traffic share count is 1
```

What happens on cust-R2 that causes it to send this traffic back to ISP-R1?

On cust-R2:

```
*Oct  6 09:41:43.495: IP: s=10.3.3.3 (Serial2/0), d=192.168.20.1 (Serial2/0), g=10.0.0.1, len 100,
*Oct  6 09:41:43.515: IP: tableid=0, s=10.3.3.3 (Serial2/0), d=192.168.20.1 (Serial2/0), routed vi
*Oct  6 09:41:43.515: IP: s=10.3.3.3 (Serial2/0), d=192.168.20.1 (Serial2/0), g=10.0.0.1, len 100,
*Oct  6 09:41:43.555: IP: tableid=0, s=10.3.3.3 (Serial2/0), d=192.168.20.1 (Serial2/0), routed vi
```

You can see that cust-R2 sends these packets back to ISP-R1, because of this route:

```
<#root>

cust-R2#

show ip route 192.168.20.1 longer-prefixes

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.0.0.1 to network 0.0.0.0

cust-R2#
```

Router cust-R2 does not have a route to 192.168.20.1 because this network is not in use in the user network, so the best route to 192.168.20.1 is the default route which points to ISP-R1.The result is that the packets loop between ISP-R1 and cust-R2 until the TTL expires.

If the ICMP request had gone to an IP address within a network that is in use, this result would not occur.

For example, if the ICMP request was for 192.168.1.x, which is directly connected on cust-R2, no looping would have occurred:

```
<#root>

cust-R2#

show ip route 192.168.1.1

Routing entry for 192.168.1.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Ethernet0/0
      Route metric is 0, traffic share count is 1
```

The solution to this problem is to configure a static route to Null0 for 192.168.0.0/16 on cust-R2.

```
<#root>

cust-R2#

configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
cust-R2(config)#

ip route 192.168.0.0 255.255.0.0 Null0

cust-R2(config)#

end

cust-R2#
*Oct  6 09:53:18.015: %SYS-5-CONFIG_I: Configured from console by console

cust-R2#

show ip route 192.168.20.1

Routing entry for 192.168.0.0/16, supernet
  Known via "static", distance 1, metric 0 (connected)
  Routing Descriptor Blocks:


 * directly connected, via Null0

      Route metric is 0, traffic share count is 1
```

If you now resend the ICMP request from BB-R3 to 192.168.20.1, cust-R2 sends this traffic to Null0, which triggers an ICMP unreachable to be generated.

```
<#root>

BB-R3#

ping ip 192.168.20.1 repeat 1


Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
U
Success rate is 0 percent (0/1)
BB-R3#
*Oct  6 09:54:33.051: ICMP: dst (10.3.3.3) host unreachable rcv from 10.0.0.2
```

There can be situations where the use of a summary static route to Null0 is not feasible. For example, if in the previous example:

- Block 192.168.1.0/24 is connected to another router which dials into cust-R2 via ISDN
- ISP-R1 does not allocate 192.168.0.0/16 but only 192.168.1.0/24
- A disconnection of the ISDN link occurs

Note: The result would be that packets in transit or applications that attempt to reach this block of IP addresses create the same routing loop described earlier.

Note: To fix this routing loop, you must use the **ip route 192.168.1.0 255.255.255.0 Null0 200** command to configure a floating static route to Null0 for 192.168.1.0/24. The 200 in the command is the administrative distance. Refer to Describe Administrative Distance for more information.

Note: Because you use a higher administrative distance than any routing protocol, if the route to 192.168.1.0/24 via the ISDN link becomes inactive, cust-R2 installs a floating static route. Packets are then sent to Null0 until the ISDN link becomes active.

Related Information

- Cisco Technical Support & Downloads