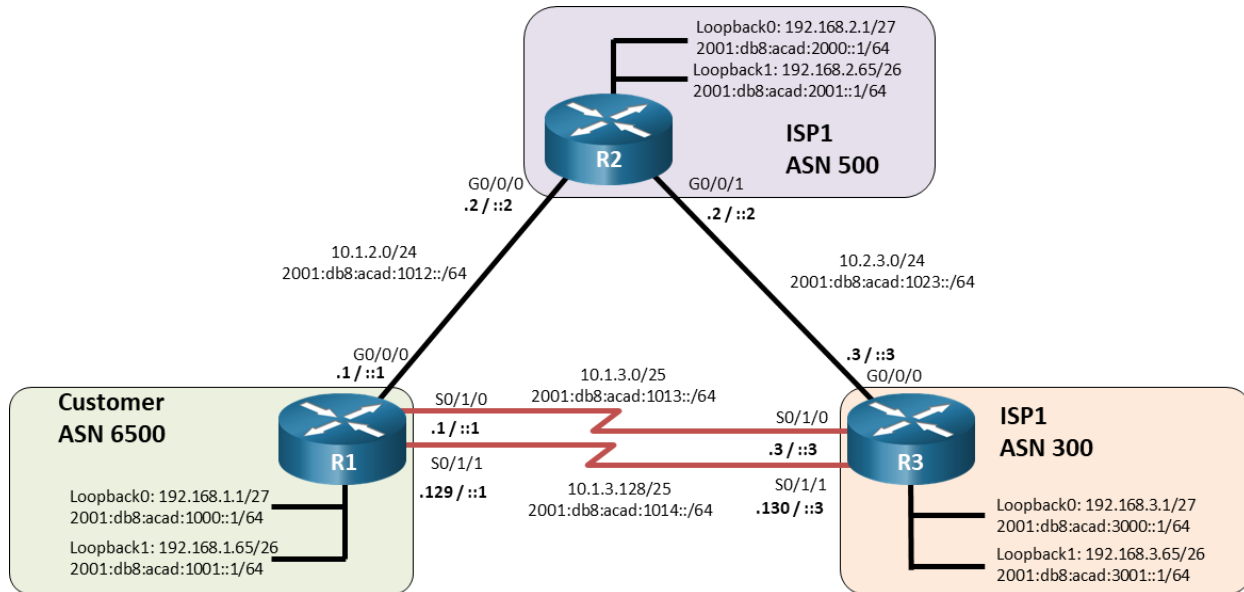


Lab - Implement BGP Path Manipulation (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Answers: [12.1.2 Lab - Implement BGP Path Manipulation](#)

Topology



Addressing Table

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
R1	G0/0/0	10.1.2.1/24	2001:db8:acad:1012::1/64	fe80::1:1
	S0/1/0	10.1.3.1/25	2001:db8:acad:1013::1/64	fe80::1:2
	S0/1/1	10.1.3.129/25	2001:db8:acad:1014::1/64	fe80::1:3
	Loopback0	192.168.1.1/27	2001:db8:acad:1000::1/64	fe80::1:4
	Loopback1	192.168.1.65/26	2001:db8:acad:1001::1/64	fe80::1:5
R2	G0/0/0	10.1.2.2/24	2001:db8:acad:1012::2/64	fe80::2:1
	G0/0/1	10.2.3.2/24	2001:db8:acad:1023::2/64	fe80::2:2
	Loopback0	192.168.2.1/27	2001:db8:acad:2000::1/64	fe80::2:4
	Loopback1	192.168.2.65/26	2001:db8:acad:2001::1/64	fe80::2:4
R3	G0/0/0	10.2.3.3/24	2001:db8:acad:1023::3/64	fe80::3:1
	S0/1/0	10.1.3.3/25	2001:db8:acad:1013::3/64	fe80::3:2
	S0/1/1	10.1.3.130/25	2001:db8:acad:1014::3/64	fe80::3:3

Device	Interface	IPv4 Address	IPv6 Address	IPv6 Link-Local
	Loopback0	192.168.3.1/27	2001:db8:acad:3000::1/64	fe80::3:4
	Loopback1	192.168.3.65/26	2001:db8:acad:3001::1/64	fe80::3:5

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Configure and Verify Multi-Protocol BGP on all Routers

Part 3: Configure and Verify BGP Path Manipulation Settings on all Routers

Background / Scenario

The default settings in BGP allow for a great deal of undesired route information to pass between autonomous systems. In this lab you will configure Multi-Protocol BGP and implement various path manipulation options for both IPv4 and IPv6.

Note: This lab is an exercise in developing, deploying, and verifying various path manipulation tools for BGP, and does not reflect networking best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 PC (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each router.

- a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing. A command list for each router is listed below to perform initial configuration.

Router R1

Lab - Implement BGP Path Manipulation

```
no ip domain lookup
hostname R1
line con 0
  exec-timeout 0 0
  logging synchronous
banner motd # This is R1, BGP Path Manipulation Lab #
ipv6 unicast-routing
interface g0/0/0
  ip address 10.1.2.1 255.255.255.0
  ipv6 address fe80::1:1 link-local
  ipv6 address 2001:db8:acad:1012::1/64
  no shutdown
interface s0/1/0
  ip address 10.1.3.1 255.255.255.128
  ipv6 address fe80::1:2 link-local
  ipv6 address 2001:db8:acad:1013::1/64
  no shutdown
interface s0/1/1
  ip address 10.1.3.129 255.255.255.128
  ipv6 address fe80::1:3 link-local
  ipv6 address 2001:db8:acad:1014::1/64
  no shutdown
interface loopback 0
  ip address 192.168.1.1 255.255.255.224
  ipv6 address fe80::1:4 link-local
  ipv6 address 2001:db8:acad:1000::1/64
  no shutdown
interface loopback 1
  ip address 192.168.1.65 255.255.255.192
  ipv6 address fe80::1:5 link-local
  ipv6 address 2001:db8:acad:1001::1/64
  no shutdown
```

Router R2

```
no ip domain lookup
hostname R2
line con 0
  exec-timeout 0 0
  logging synchronous
banner motd # This is R2, BGP Path Manipulation Lab #
ipv6 unicast-routing
interface g0/0/0
  ip address 10.1.2.2 255.255.255.0
  ipv6 address fe80::2:1 link-local
  ipv6 address 2001:db8:acad:1012::2/64
  no shutdown
```

Lab - Implement BGP Path Manipulation

```
interface g0/0/1
 ip address 10.2.3.2 255.255.255.0
 ipv6 address fe80::2:2 link-local
 ipv6 address 2001:db8:acad:1023::2/64
 no shutdown
interface loopback 0
 ip address 192.168.2.1 255.255.255.224
 ipv6 address fe80::2:3 link-local
 ipv6 address 2001:db8:acad:2000::1/64
 no shutdown
interface loopback 1
 ip address 192.168.2.65 255.255.255.192
 ipv6 address fe80::2:4 link-local
 ipv6 address 2001:db8:acad:2001::1/64
 no shutdown
```

Router R3

```
no ip domain lookup
hostname R3
line con 0
 exec-timeout 0 0
 logging synchronous
banner motd # This is R3, BGP Path Manipulation Lab #
ipv6 unicast-routing
interface g0/0/0
 ip address 10.2.3.3 255.255.255.0
 ipv6 address fe80::3:1 link-local
 ipv6 address 2001:db8:acad:1023::3/64
 no shutdown
interface s0/1/0
 ip address 10.1.3.3 255.255.255.128
 ipv6 address fe80::3:2 link-local
 ipv6 address 2001:db8:acad:1013::3/64
 no shutdown
interface s0/1/1
 ip address 10.1.3.130 255.255.255.128
 ipv6 address fe80::3:3 link-local
 ipv6 address 2001:db8:acad:1014::3/64
 no shutdown
interface loopback 0
 ip address 192.168.3.1 255.255.255.224
 ipv6 address fe80::3:4 link-local
 ipv6 address 2001:db8:acad:3000::1/64
 no shutdown
interface loopback 1
 ip address 192.168.3.65 255.255.255.192
```

```
ipv6 address fe80::3:5 link-local
ipv6 address 2001:db8:acad:3001::1/64
no shutdown
```

- b. Set the clock on each router to UTC time.
- c. Save the running configuration to startup-config.

Part 2: Configure and Verify Multi-Protocol BGP on all Routers

In Part 2, you will configure and verify Multi-Protocol BGP on all routers to achieve full connectivity between the routers. The text below provides you with the complete configuration for R1. You will use this to inform your configuration of R2 and R3. The configuration being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: On R1, create the core BGP configuration.

- a. Enter BGP configuration mode from global configuration mode, specifying AS 6500.

```
R1(config)# router bgp 6500
```

- b. Configure the BGP router-id for R1.

```
R1(config-router)# bgp router-id 1.1.1.1
```

- c. Disable the default IPv4 unicast address family behavior.

```
R1(config-router)# no bgp default ipv4-unicast
```

- d. Based on the topology diagram, configure all the designated neighbors for R1.

```
R1(config-router)# neighbor 10.1.2.2 remote-as 500
```

```
R1(config-router)# neighbor 10.1.3.3 remote-as 300
```

```
R1(config-router)# neighbor 10.1.3.130 remote-as 300
```

```
R1(config-router)# neighbor 2001:db8:acad:1012::2 remote-as 500
```

```
R1(config-router)# neighbor 2001:db8:acad:1013::3 remote-as 300
```

```
R1(config-router)# neighbor 2001:db8:acad:1014::3 remote-as 300
```

Step 2: On R1, configure the IPv4 unicast address family.

- a. Enter the IPv4 unicast address family configuration mode.

```
R1(config-router)# address-family ipv4 unicast
```

- b. Configure network statements for the IPv4 networks attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does, and that the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.

```
R1(config-router-af)# network 192.168.1.0 mask 255.255.255.224
```

```
R1(config-router-af)# network 192.168.1.64 mask 255.255.255.192
```

- c. Deactivate the IPv6 neighbors and activate the IPv4 neighbors.

```
R1(config-router-af)# no neighbor 2001:db8:acad:1012::2 activate
```

```
R1(config-router-af)# no neighbor 2001:db8:acad:1013::3 activate
```

```
R1(config-router-af)# no neighbor 2001:db8:acad:1014::3 activate
```

```
R1(config-router-af)# neighbor 10.1.2.2 activate
```

```
R1(config-router-af)# neighbor 10.1.3.3 activate
```

```
R1(config-router-af)# neighbor 10.1.3.130 activate
```

Step 3: On R1, configure the IPv6 unicast address family.

- a. Enter the IPv6 unicast address family configuration mode.
R1(config-router)# **address-family ipv6 unicast**
- b. Configure network statements for the IPv6 networks that are attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does; therefore, the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.
R1(config-router-af)# **network 2001:db8:acad:1000::/64**
R1(config-router-af)# **network 2001:db8:acad:1001::/64**
- c. Activate the IPv6 neighbors that are configured for BGP.
R1(config-router-af)# **neighbor 2001:db8:acad:1012::2 activate**
R1(config-router-af)# **neighbor 2001:db8:acad:1013::3 activate**
R1(config-router-af)# **neighbor 2001:db8:acad:1014::3 activate**

Step 4: Configure MP-BGP on R2 and R3 as you did in the previous step.

```
R2(config)# router bgp 500
R2(config-router)# bgp router-id 2.2.2.2
R2(config-router)# no bgp default ipv4-unicast
R2(config-router)# neighbor 10.1.2.1 remote-as 6500
R2(config-router)# neighbor 10.2.3.3 remote-as 300
R2(config-router)# neighbor 2001:db8:acad:1012::1 remote-as 6500
R2(config-router)# neighbor 2001:db8:acad:1023::3 remote-as 300
R2(config-router)# address-family ipv4
R2(config-router-af)# network 192.168.2.0 mask 255.255.255.224
R2(config-router-af)# network 192.168.2.64 mask 255.255.255.192
R2(config-router-af)# neighbor 10.1.2.1 activate
R2(config-router-af)# neighbor 10.2.3.3 activate
R2(config-router-af)# no neighbor 2001:db8:acad:1012::1 activate
R2(config-router-af)# no neighbor 2001:db8:acad:1023::3 activate
R2(config-router-af)# exit-address-family
R2(config-router)# address-family ipv6
R2(config-router-af)# network 2001:db8:acad:2000::/64
R2(config-router-af)# network 2001:db8:acad:2001::/64
R2(config-router-af)# neighbor 2001:db8:acad:1012::1 activate
R2(config-router-af)# neighbor 2001:db8:acad:1023::3 activate
R2(config-router-af)# exit-address-family

R3(config)# router bgp 300
R3(config-router)# bgp router-id 3.3.3.3
R3(config-router)# no bgp default ipv4-unicast
R3(config-router)# neighbor 10.1.3.1 remote-as 6500
R3(config-router)# neighbor 10.1.3.129 remote-as 6500
R3(config-router)# neighbor 10.2.3.2 remote-as 500
R3(config-router)# neighbor 2001:db8:acad:1013::1 remote-as 6500
R3(config-router)# neighbor 2001:db8:acad:1014::1 remote-as 6500
R3(config-router)# neighbor 2001:db8:acad:1023::2 remote-as 500
```

Lab - Implement BGP Path Manipulation

```
R3(config-router)# address-family ipv4
R3(config-router-af)# network 192.168.3.0 mask 255.255.255.224
R3(config-router-af)# network 192.168.3.64 mask 255.255.255.192
R3(config-router-af)# neighbor 10.1.3.1 activate
R3(config-router-af)# neighbor 10.1.3.129 activate
R3(config-router-af)# neighbor 10.2.3.2 activate
R3(config-router-af)# no neighbor 2001:db8:acad:1013::1 activate
R3(config-router-af)# no neighbor 2001:db8:acad:1014::1 activate
R3(config-router-af)# no neighbor 2001:db8:acad:1023::2 activate
R3(config-router-af)# exit-address-family
R3(config-router)# address-family ipv6
R3(config-router-af)# network 2001:db8:acad:3000::/64
R3(config-router-af)# network 2001:db8:acad:3001::/64
R3(config-router-af)# neighbor 2001:db8:acad:1013::1 activate
R3(config-router-af)# neighbor 2001:db8:acad:1014::1 activate
R3(config-router-af)# neighbor 2001:db8:acad:1023::2 activate
R3(config-router-af)# exit-address-family
```

Step 5: Verify that MP-BGP is operational.

- Use the **show bgp ipv4 unicast summary** and **show bgp ipv6 unicast summary** commands to verify that BGP has established three IPv4 and three IPv6 adjacencies and received four prefixes from each neighbor.

```
R1# show bgp ipv4 unicast summary
```

```
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 9, main routing table version 9
6 network entries using 1488 bytes of memory
14 path entries using 1904 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4920 total bytes of memory
BGP activity 12/0 prefixes, 28/0 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.1.2.2	4	500	8	8	9	0	0	00:02:42	4
10.1.3.3	4	300	8	8	9	0	0	00:02:12	4
10.1.3.130	4	300	8	8	9	0	0	00:02:11	4

```
R1# show bgp ipv6 unicast summary
```

```
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 9, main routing table version 9
6 network entries using 1632 bytes of memory
14 path entries using 2128 bytes of memory
5/3 BGP path/bestpath attribute entries using 1400 bytes of memory
4 BGP AS-PATH entries using 128 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
```

Lab - Implement BGP Path Manipulation

BGP using 5288 total bytes of memory
 BGP activity 12/0 prefixes, 28/0 paths, scan interval 60 secs

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2001:DB8:ACAD:1012::2	4	500	8	8	9	0	0	00:02:50	4
2001:DB8:ACAD:1013::3	4	300	8	8	9	0	0	00:02:14	4
2001:DB8:ACAD:1014::3	4	300	8	8	9	0	0	00:02:13	4

- b. Use the **show bgp ipv4 unicast** and **show bgp ipv6 unicast** commands to view the specified BGP tables. Note that R1 has multiple paths to each destination network. Take note of the next hop address for the destination networks marked with the ">" symbol.

R1# **show bgp ipv4 unicast | begin Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0/27	0.0.0.0	0		32768	i
*> 192.168.1.64/26	0.0.0.0	0		32768	i
* 192.168.2.0/27	10.1.3.130			0	300 500 i
*>	10.1.2.2	0		0	500 i
*	10.1.3.3			0	300 500 i
* 192.168.2.64/26	10.1.3.130			0	300 500 i
*>	10.1.2.2	0		0	500 i
*	10.1.3.3			0	300 500 i
* 192.168.3.0/27	10.1.3.130			0	300 i
*	10.1.2.2			0	500 300 i
*>	10.1.3.3	0		0	300 i
* 192.168.3.64/26	10.1.3.130			0	300 i
*	10.1.2.2			0	500 300 i
*>	10.1.3.3	0		0	300 i

R1# **show bgp ipv6 unicast | begin Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:DB8:ACAD:1000::/64	::	0		32768	i
*> 2001:DB8:ACAD:1001::/64	::	0		32768	i
* 2001:DB8:ACAD:2000::/64	2001:DB8:ACAD:1013::3			0	300 500 i
*>	2001:DB8:ACAD:1012::2	0		0	500 i
*	2001:DB8:ACAD:1014::3			0	300 500 i
* 2001:DB8:ACAD:2001::/64	2001:DB8:ACAD:1013::3			0	300 500 i
*>	2001:DB8:ACAD:1012::2	0		0	500 i
*	2001:DB8:ACAD:1014::3			0	300 500 i

Lab - Implement BGP Path Manipulation

```
0 300 500 i
*> 2001:DB8:ACAD:3000::/64
      2001:DB8:ACAD:1013::3
      0 0 300 i
*      2001:DB8:ACAD:1012::2
      0 500 300 i
*      2001:DB8:ACAD:1014::3
      0 0 300 i
*> 2001:DB8:ACAD:3001::/64
      2001:DB8:ACAD:1013::3
      0 0 300 i
*      2001:DB8:ACAD:1012::2
      0 500 300 i
*      2001:DB8:ACAD:1014::3
      0 0 300 i
```

- c. Use the **show ip route bgp** and **show ipv6 route bgp** commands to view the routing tables. Note that there is only one route to each destination, and that the routes included in the routing table have the same next hop as those with the ">" symbol in the BGP tables.

```
R1# show ip route bgp | begin Gateway
```

```
Gateway of last resort is not set
```

```
192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B 192.168.2.0/27 [20/0] via 10.1.2.2, 00:04:10
B 192.168.2.64/26 [20/0] via 10.1.2.2, 00:04:10
192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B 192.168.3.0/27 [20/0] via 10.1.3.3, 00:04:09
B 192.168.3.64/26 [20/0] via 10.1.3.3, 00:04:09
```

```
R1# show ipv6 route bgp
```

```
IPv6 Routing Table - default - 15 entries
```

```
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
```

```
B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
```

```
I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
```

```
EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
```

```
NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
```

```
OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
```

```
ON2 - OSPF NSSA ext 2, a - Application
```

```
B 2001:DB8:ACAD:2000::/64 [20/0]
   via FE80::2:1, GigabitEthernet0/0/0
B 2001:DB8:ACAD:2001::/64 [20/0]
   via FE80::2:1, GigabitEthernet0/0/0
B 2001:DB8:ACAD:3000::/64 [20/0]
   via FE80::3:2, Serial0/1/0
B 2001:DB8:ACAD:3001::/64 [20/0]
   via FE80::3:2, Serial0/1/0
```

Part 3: Configure and Verify BGP Path Manipulation Settings on all Routers

In Part 3, you will configure path manipulation tools for BGP. The way these tools are being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: Configure ACL-based route filtering.

In this step, you will configure R3 so that it only sends ASN300 networks to R1; it will not tell R1 that it knows about the networks in ASN200.

- On R1, issue the command **show bgp ipv4 unicast | i 300** to see what prefixes ASN300 is sharing via BGP. Take note of those prefixes that do not originate in ASN300.

```
R1# show bgp ipv4 unicast | i 300
* 192.168.2.0/27 10.1.3.3 0 300 500 i
* 10.1.3.130 0 300 500 i
* 192.168.2.64/26 10.1.3.3 0 300 500 i
* 10.1.3.130 0 300 500 i
* 192.168.3.0/27 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 300 i
* 10.1.3.130 0 300 i
* 192.168.3.64/26 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 300 i
* 10.1.3.130 0 300 i
```

- On R3, configure an access list designed to match the source address and mask of the networks belonging to ASN300:

```
R3(config)# ip access-list extended ALLOWED_TO_R1
R3(config-ext-nacl)# permit ip 192.168.3.0 0.0.0.0 255.255.255.224 0.0.0.0
R3(config-ext-nacl)# permit ip 192.168.3.64 0.0.0.0 255.255.255.192 0.0.0.0
R3(config-ext-nacl)# exit
```

- On R3, apply the ALLOWED_TO_R1 ACL as a distribute list to the IPv4 neighbor adjacencies with R1.

```
R3(config)# router bgp 300
R3(config-router)# address-family ipv4 unicast
R3(config-router-af)# neighbor 10.1.3.1 distribute-list ALLOWED_TO_R1 out
R3(config-router-af)# neighbor 10.1.3.129 distribute-list ALLOWED_TO_R1 out
R3(config-router-af)# end
```

- Perform a reset of the IPv4 adjacency with R1 for the outbound traffic without tearing down the session.

```
R3# clear bgp ipv4 unicast 6500 out
```

- On R1, issue the command **show bgp ipv4 unicast | i 300** to see what prefixes routes ASN300 is now sharing via BGP. All of the prefixes should now originate in ASN300:

```
R1# show bgp ipv4 unicast | i 300
* 192.168.3.0/27 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 300 i
* 10.1.3.130 0 300 i
* 192.168.3.64/26 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 300 i
* 10.1.3.130 0 300 i
```

Step 2: Configure prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 networks from R2; it will not accept information about ASN300 networks from R2.

- On R1, issue the command **show bgp ipv4 unicast | begin 192.168.3** to see what prefixes ASN500 is sharing via BGP. Take note of those prefixes that do not originate in ASN500.

Lab - Implement BGP Path Manipulation

```
R1# show bgp ipv4 unicast | begin 192.168.3
* 192.168.3.0/27 10.1.3.130 0 0 300 i
* 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 0 300 i
* 192.168.3.64/26 10.1.3.130 0 0 300 i
* 10.1.2.2 0 500 300 i
*> 10.1.3.3 0 0 300 i
```

- b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN500.

```
R1(config)# ip prefix-list ALLOWED_FROM_R2 seq 5 permit 192.168.2.0/24 le 27
```

- c. Apply the ALLOWED_FROM_R2 prefix list to the IPv4 neighbor adjacencies for R2.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.2.2 prefix-list ALLOWED_FROM_R2 in
R1(config-router-af)# end
```

- d. Perform a reset of the IPv4 adjacency with R2 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 500 in
```

- e. On R1, issue the command **show bgp ipv4 unicast | i 500** to see what prefixes routes ASN500 is now sharing via BGP. All of the prefixes should now originate in ASN500.

```
R1# show bgp ipv4 unicast | i 500
*> 192.168.2.0/27 10.1.2.2 0 0 500 i
*> 192.168.2.64/26 10.1.2.2 0 0 500 i
```

Step 3: Configure an AS-PATH ACL to filter routes being advertised.

In this step, you will configure R1 so that it only sends ASN100 networks to R2; it will not forward information about prefixes from any other ASN to ASN500.

- a. On R2, issue the command **show bgp ipv4 unicast | begin Network** to see what prefixes ASN6500 is sharing via BGP. Take note of those prefixes that do not originate in ASN6500. Advertising these routes could set ASN6500 up as a transit AS, and that is not a desirable scenario.

```
R2# show bgp ipv4 unicast | begin Network
      Network          Next Hop          Metric LocPrf Weight Path
* 192.168.1.0/27      10.2.3.3
*> 10.1.2.1 0 0 6500 i
* 192.168.1.64/26    10.2.3.3
*> 10.1.2.1 0 0 6500 i
*> 192.168.2.0/27    0.0.0.0 0 32768 i
*> 192.168.2.64/26  0.0.0.0 0 32768 i
* 192.168.3.0/27    10.1.2.1 0 6500 300 i
*> 10.2.3.3 0 0 300 i
* 192.168.3.64/26   10.1.2.1 0 6500 300 i
*> 10.2.3.3 0 0 300 i
```

- b. On R1, configure AS-PATH ACL to match the routes from the local ASN.

```
R1(config)# ip as-path access-list 1 permit ^$
```

- c. On R1, apply the AS-PATH ACL as a filter-list on the adjacency configured with R2.

```
R1(config)# router bgp 6500
```

Lab - Implement BGP Path Manipulation

```
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.2.2 filter-list 1 out
R1(config-router-af)# end
```

- d. On R1, perform a reset of the IPv4 adjacency with R2 for the outbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 500 out
```

- e. On R2, issue the command **show bgp ipv4 unicast | i 6500** to see what prefixes routes ASN6500 is now sharing via BGP. All of the prefixes should now originate in ASN6500.

```
R2# show bgp ipv4 unicast | i 6500
*   192.168.1.0/27   10.2.3.3           0 300 6500 i
*>
*   192.168.1.64/26 10.2.3.3           0 300 6500 i
*>
```

Step 4: Configure IPv6 prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 IPv6 networks from R2. It will not accept information about ASN300 IPv6 networks from R2.

- a. On R1, issue the command **show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes** to see what IPv6 prefixes ASN500 is sharing via BGP. Take note of those IPv6 prefixes that do not originate in ASN500.

```
R1# show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes
BGP table version is 9, local router ID is 1.1.1.1
Status code001s: s suppressed, d damped, h history, * valid, > best, i - internal,
                 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
                 x best-external, a additional-path, c RIB-compressed,
                 t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	2001:DB8:ACAD:2000::/64	2001:DB8:ACAD:1012::2	0		500	i
*>	2001:DB8:ACAD:2001::/64	2001:DB8:ACAD:1012::2	0		500	i
*	2001:DB8:ACAD:3000::/64	2001:DB8:ACAD:1012::2			500	300 i
*	2001:DB8:ACAD:3001::/64	2001:DB8:ACAD:1012::2			500	300 i

```
Total number of prefixes 4
```

- b. On R1, configure an IPv6 prefix list designed to match the source address and mask of networks belonging to ASN500.

Lab - Implement BGP Path Manipulation

```
R1(config)# ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 5 permit
2001:db8:acad:2000::/64
R1(config)# ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 10 permit
2001:db8:acad:2001::/64
```

- c. Apply the IPV6_ALLOWED_FROM_R2 prefix list to the IPv6 neighbor adjacencies for R2.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv6 unicast
R1(config-router-af)# neighbor 2001:db8:acad:1012::2 prefix-list
IPV6_ALLOWED_FROM_R2 in
R1(config-router-af)# end
```

- d. Perform a reset of the IPv6 adjacency with R2 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv6 unicast 500 in
```

- e. On R1, issue the command **show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes** to see what IPv6 prefixes routes ASN500 is now sharing via BGP. All of the IPv6 prefixes should now originate in ASN500.

```
R1# show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:DB8:ACAD:2000::/64	2001:DB8:ACAD:1012::2	0		0	500 i
*> 2001:DB8:ACAD:2001::/64	2001:DB8:ACAD:1012::2	0		0	500 i

```
Total number of prefixes 2
```

- f. Configure and apply an IPv6 filter to do the same thing on the adjacency with ASN300.

Step 5: Configure BGP path attribute manipulation to effect routing.

In this step, you will configure R1 so that it prefers the next-hop address of 192.168.3.130 over 192.168.3.3, which would normally be the preferred path to ASN300 networks. You will do this by using a prefix list to identify the destination networks and then use a route map to match the prefix list and set the matched networks to have a local preference of 250.

- a. On R1, issue the command **show ip route bgp** and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks. Then issue the command **show bgp ipv4 unicast** and note that the 10.1.3.130 is a valid next hop (It's just not the *best* next hop, according to the BGP path selection algorithm.) Lastly, issue the command **show bgp ipv4 unicast 192.168.3.0** to see details about all the paths available and which one was selected.

```
R1# show bgp ipv4 unicast 192.168.3.0
BGP routing table entry for 192.168.3.0/27, version 8
```

Lab - Implement BGP Path Manipulation

```
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  300
  10.1.3.3 from 10.1.3.3 (3.3.3.3)
    Origin IGP, metric 0, localpref 100, valid, external, best
    rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  300
  10.1.3.130 from 10.1.3.130 (3.3.3.3)
    Origin IGP, metric 0, localpref 100, valid, external
    rx pathid: 0, tx pathid: 0
```

- b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN300.

```
R1(config)# ip prefix-list PREFERRED_IPV4_PATH seq 5 permit 192.168.3.0/24 le 27
```

- c. Create a route-map named USE_THIS_PATH_FOR_IPV4 that matches on the prefix list you just created and sets the local preference to 250.

```
R1(config)# route-map USE_THIS_PATH_FOR_IPV4 permit 10
R1(config)# match ip address prefix-list PREFERRED_IPV4_PATH
R1(config)# set local-preference 250
```

- d. Next, apply this route map to the BGP neighbor 10.1.3.130.

```
R1(config)# router bgp 6500
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.1.3.130 route-map USE_THIS_PATH_FOR_IPV4 in
R1(config-router-af)# end
```

- e. Perform a reset of the IPv4 adjacency with R3 for the inbound traffic without tearing down the session.

```
R1# clear bgp ipv4 unicast 300 in
```

- f. On R1, issue the command **show ip route bgp** and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks; it should be 10.1.3.130 for both. Issue the command **show bgp ipv4 unicast** and you should see the local preference value in the appropriate column.

```
R1# show ip route bgp | begin Gateway
```

```
Gateway of last resort is not set
```

```
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.2.0/27 [20/0] via 10.1.2.2, 00:35:17
B       192.168.2.64/26 [20/0] via 10.1.2.2, 00:35:17
    192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.3.0/27 [20/0] via 10.1.3.130, 00:00:08
B       192.168.3.64/26 [20/0] via 10.1.3.130, 00:00:08
```

```
R1# show bgp ipv4 unicast | begin Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.168.1.0/27	0.0.0.0		0	32768	i
*>	192.168.1.64/26	0.0.0.0		0	32768	i
*>	192.168.2.0/27	10.1.2.2		0		0 500 i

Lab - Implement BGP Path Manipulation

```
*> 192.168.2.64/26 10.1.2.2 0 0 500 i
* 192.168.3.0/27 10.1.3.3 0 0 300 i
*> 10.1.3.130 0 250 0 300 i
* 192.168.3.64/26 10.1.3.3 0 0 300 i
*> 10.1.3.130 0 250 0 300 i
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

Device Configs - Final

Router R1

```
R1# show run
```

```
Building configuration...
```

```
Current configuration : 5819 bytes
```

```
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R1
!
```

Lab - Implement BGP Path Manipulation

```
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
ipv6 unicast-routing
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
redundancy
mode none
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.224
 ipv6 address FE80::1:4 link-local
 ipv6 address 2001:DB8:ACAD:1000::1/64
!
interface Loopback1
 ip address 192.168.1.65 255.255.255.192
 ipv6 address FE80::1:5 link-local
 ipv6 address 2001:DB8:ACAD:1001::1/64
!
interface GigabitEthernet0/0/0
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
 ipv6 address FE80::1:1 link-local
 ipv6 address 2001:DB8:ACAD:1012::1/64
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
interface Serial0/1/0
 ip address 10.1.3.1 255.255.255.128
 ipv6 address FE80::1:2 link-local
 ipv6 address 2001:DB8:ACAD:1013::1/64
!
interface Serial0/1/1
 ip address 10.1.3.129 255.255.255.128
 ipv6 address FE80::1:3 link-local
 ipv6 address 2001:DB8:ACAD:1014::1/64
!
router bgp 6500
```


Lab - Implement BGP Path Manipulation

```
bgp router-id 1.1.1.1
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.1.2.2 remote-as 500
neighbor 10.1.3.3 remote-as 300
neighbor 10.1.3.130 remote-as 300
neighbor 2001:DB8:ACAD:1012::2 remote-as 500
neighbor 2001:DB8:ACAD:1013::3 remote-as 300
neighbor 2001:DB8:ACAD:1014::3 remote-as 300
!
address-family ipv4
  network 192.168.1.0 mask 255.255.255.224
  network 192.168.1.64 mask 255.255.255.192
  neighbor 10.1.2.2 activate
  neighbor 10.1.2.2 prefix-list ALLOWED_FROM_R2 in
  neighbor 10.1.2.2 filter-list 1 out
  neighbor 10.1.3.3 activate
  neighbor 10.1.3.130 activate
  neighbor 10.1.3.130 route-map USE_THIS_PATH_FOR_IPV4 in
exit-address-family
!
address-family ipv6
  network 2001:DB8:ACAD:1000::/64
  network 2001:DB8:ACAD:1001::/64
  neighbor 2001:DB8:ACAD:1012::2 activate
  neighbor 2001:DB8:ACAD:1012::2 prefix-list IPV6_ALLOWED_FROM_R2 in
  neighbor 2001:DB8:ACAD:1013::3 activate
  neighbor 2001:DB8:ACAD:1014::3 activate
exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
ip as-path access-list 1 permit ^$
!
ip prefix-list ALLOWED_FROM_R2 seq 5 permit 192.168.2.0/24 le 27
!
ip prefix-list PREFERRED_IPV4_PATH seq 5 permit 192.168.3.0/24 le 27
!
ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 5 permit 2001:DB8:ACAD:2000::/64
ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 10 permit 2001:DB8:ACAD:2001::/64
!
route-map USE_THIS_PATH_FOR_IPV4 permit 10
  match ip address prefix-list PREFERRED_IPV4_PATH
  set local-preference 250
!
control-plane
!
banner motd ^C This is R1, BGP Path Manipulation Lab ^C
```

Lab - Implement BGP Path Manipulation

```
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
  transport input none  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4  
  login  
!  
end
```

Router R2

```
R2# show run  
Building configuration...  
  
Current configuration : 4600 bytes  
!  
version 16.9  
service timestamps debug datetime msec  
service timestamps log datetime msec  
platform qfp utilization monitor load 80  
no platform punt-keepalive disable-kernel-core  
!  
hostname R2  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
!  
no ip domain lookup  
!  
ip dhcp pool webuidhcp  
!  
login on-success log  
!  
subscriber templating  
!  
ipv6 unicast-routing  
multilink bundle-name authenticated  
!  
spanning-tree extend system-id  
!  
redundancy  
  mode none  
!  
interface Loopback0
```

Lab - Implement BGP Path Manipulation

```
ip address 192.168.2.1 255.255.255.224
ipv6 address FE80::2:3 link-local
ipv6 address 2001:DB8:ACAD:2000::1/64
!
interface Loopback1
ip address 192.168.2.65 255.255.255.192
ipv6 address FE80::2:4 link-local
ipv6 address 2001:DB8:ACAD:2001::1/64
!
interface GigabitEthernet0/0/0
ip address 10.1.2.2 255.255.255.0
negotiation auto
ipv6 address FE80::2:1 link-local
ipv6 address 2001:DB8:ACAD:1012::2/64
!
interface GigabitEthernet0/0/1
ip address 10.2.3.2 255.255.255.0
negotiation auto
ipv6 address FE80::2:2 link-local
ipv6 address 2001:DB8:ACAD:1023::2/64
!
router bgp 500
bgp router-id 2.2.2.2
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.1.2.1 remote-as 6500
neighbor 10.2.3.3 remote-as 300
neighbor 2001:DB8:ACAD:1012::1 remote-as 6500
neighbor 2001:DB8:ACAD:1023::3 remote-as 300
!
address-family ipv4
network 192.168.2.0 mask 255.255.255.224
network 192.168.2.64 mask 255.255.255.192
neighbor 10.1.2.1 activate
neighbor 10.2.3.3 activate
exit-address-family
!
address-family ipv6
network 2001:DB8:ACAD:2000::/64
network 2001:DB8:ACAD:2001::/64
neighbor 2001:DB8:ACAD:1012::1 activate
neighbor 2001:DB8:ACAD:1023::3 activate
exit-address-family
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
control-plane
!
```

Lab - Implement BGP Path Manipulation

```
banner motd ^C This is R2, BGP Path Manipulation Lab ^C
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
  transport input none  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4  
  login  
!  
end
```

Router R3

```
R3# show run  
Building configuration...  
  
Current configuration : 5180 bytes  
!  
version 16.9  
service timestamps debug datetime msec  
service timestamps log datetime msec  
platform qfp utilization monitor load 80  
no platform punt-keepalive disable-kernel-core  
!  
hostname R3  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
!  
no ip domain lookup  
!  
ip dhcp po 01 webuidhcp  
!  
login on-success log  
!  
subscriber templating  
!  
ipv6 unicast-routing  
multilink bundle-name authenticated  
!  
spanning-tree extend system-id  
!  
redundancy  
  mode none  
!  
!
```

Lab - Implement BGP Path Manipulation

```
interface Loopback0
ip address 192.168.3.1 255.255.255.224
ipv6 address FE80::3:4 link-local
ipv6 address 2001:DB8:ACAD:3000::1/64
!
interface Loopback1
ip address 192.168.3.65 255.255.255.192
ipv6 address FE80::3:5 link-local
ipv6 address 2001:DB8:ACAD:3001::1/64
!
interface GigabitEthernet0/0/0
ip address 10.2.3.3 255.255.255.0
negotiation auto
ipv6 address FE80::3:1 link-local
ipv6 address 2001:DB8:ACAD:1023::3/64
!
interface GigabitEthernet0/0/1
no ip address
negotiation auto
!
interface Serial0/1/0
ip address 10.1.3.3 255.255.255.128
ipv6 address FE80::3:2 link-local
ipv6 address 2001:DB8:ACAD:1013::3/64
!
interface Serial0/1/1
ip address 10.1.3.130 255.255.255.128
ipv6 address FE80::3:3 link-local
ipv6 address 2001:DB8:ACAD:1014::3/64
!
router bgp 300
bgp router-id 3.3.3.3
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.1.3.1 remote-as 6500
neighbor 10.1.3.129 remote-as 6500
neighbor 10.2.3.2 remote-as 500
neighbor 2001:DB8:ACAD:1013::1 remote-as 6500
neighbor 2001:DB8:ACAD:1014::1 remote-as 6500
neighbor 2001:DB8:ACAD:1023::2 remote-as 500
!
address-family ipv4
network 192.168.3.0 mask 255.255.255.224
network 192.168.3.64 mask 255.255.255.192
neighbor 10.1.3.1 activate
neighbor 10.1.3.1 distribute-list ALLOWED_TO_R1 out
neighbor 10.1.3.129 activate
neighbor 10.1.3.129 distribute-list ALLOWED_TO_R1 out
neighbor 10.2.3.2 activate
exit-address-family
```

Lab - Implement BGP Path Manipulation

```
!  
address-family ipv6  
  network 2001:DB8:ACAD:3000::/64  
  network 2001:DB8:ACAD:3001::/64  
  neighbor 2001:DB8:ACAD:1013::1 activate  
  neighbor 2001:DB8:ACAD:1014::1 activate  
  neighbor 2001:DB8:ACAD:1023::2 activate  
exit-address-family  
!  
ip forward-protocol nd  
no ip http server  
ip http secure-server  
!  
ip access-list extended ALLOWED_TO_R1  
  permit ip host 192.168.3.0 host 255.255.255.224  
  permit ip host 192.168.3.64 host 255.255.255.192  
!  
control-plane  
!  
banner motd ^C This is R3, BGP Path Manipulation Lab ^C  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
  transport input none  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4  
  login  
!  
end
```