BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature introduces a lightweight security mechanism to protect external Border Gateway Protocol (eBGP) peering sessions from CPU utilization-based attacks using forged IP packets. Enabling this feature prevents attempts to hijack the eBGP peering session by a host on a network segment that is not part of either BGP network or by a host on a network segment that is not between the eBGP peers.

You enable this feature by configuring a minimum Time To Live (TTL) value for incoming IP packets received from a specific eBGP peer. When this feature is enabled, BGP will establish and maintain the session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. If the value is less than the configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This feature is both effective and easy to deploy.

Feature History for the BGP Support for TTL Security Check Feature

Release	Modification
12.0(27)S	This feature was introduced.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.
12.2(18)SXE	This feature was integrated into Cisco IOS Release 12.2(18)SXE.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for BGP Support for TTL Security Check

- BGP must be configured in your network and eBGP peering sessions must be established.
- This feature needs to be configured on each participating router. It protects the eBGP peering session in the incoming direction only and has no effect on outgoing IP packets or the remote router.

Restrictions for BGP Support for TTL Security Check

- This feature is designed to protect only eBGP peering sessions and is not supported for internal BGP (iBGP) peers and iBGP peer groups.
- When configuring the BGP Support for TTL Security Check feature to support an existing multihop peering session, you must first disable the **neighbor ebgp-multihop** router configuration command by entering the **no neighbor ebgp-multihop** command before configuring this feature with the **neighbor ttl-security** router configuration command. These commands are mutually exclusive, and only one command is required to establish a multihop peering session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.
- The effectiveness of this feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected peering sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside your network. This restriction also includes BGP peers that are not part of the local or external BGP network but are connected to the network segment between the BGP peers (for example, a switch or hub that is used to connect the local and external BGP networks).
- This feature does not protect the integrity of data sent between eBGP peers and does not validate eBGP peers through any authentication method. This feature validates only the locally configured TTL count against the TTL field in the IP packet header.

Information About BGP Support for TTL Security Check

To configure the BGP Support for TTL Security Check feature, you must understand the following concepts:

BGP Support for TTL Security Check Feature Overview

The BGP Support for TTL Security Check feature introduces a lightweight security mechanism to protect eBGP peering sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses.

This feature protects the eBGP peering session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP peering session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no ICMP message is generated. This is designed behavior; a response to a forged packet is unnecessary.

Although it is possible to forge the TTL field in an IP packet header, accurately forging the TTL count to match the TTL count from a trusted peer is impossible unless the network to which the trusted peer belongs has been compromised.

This feature supports both directly connected peering sessions and multihop eBGP peering sessions. The BGP peering session is not affected by incoming packets that contain invalid TTL values. The BGP peering session will remain open, and the router will silently discard the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

Configuring the TTL Security Check for BGP Peering Sessions

The BGP Support for TTL Security Check feature is configured with the **neighbor ttl-security** command in router configuration mode or address family configuration mode. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. The *hop-count* argument is used to configure the maximum number of hops that separate the two peers. The TTL value is determined by the router from the configured hop count. The value for this argument is a number from 1 to 254.

Configuring the TTL Security Check for Multihop BGP Peering Sessions

The BGP Support for TTL Security Check feature supports both directly connected peering sessions and multihop peering sessions. When this feature is configured for a multihop peering session, the **neighbor ebgp-multihop** router configuration command cannot be configured and is not needed to establish the peering session. These commands are mutually exclusive, and only one command is required to establish a multihop peering session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.

To configure this feature for an existing multihop session, you must first disable the existing peering session with the **no neighbor ebgp-multihop** command. The multihop peering session will be restored when you enable this feature with the **neighbor ttl-security** command.

This feature should be configured on each participating router. To maximize the effectiveness of this feature, the *hop-count* argument should be strictly configured to match the number of hops between the local and external network. However, you should also consider path variation when configuring this feature for a multihop peering session.

Benefits of the BGP Support for TTL Security Check Feature

The BGP Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect eBGP peering sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP autonomous system.

How to Secure BGP Sessions with the BGP Support for TTL Security Check Feature

This section contains the following procedures:

- Configuring the TTL-Security Check, page 4 (required)
- Verifying the TTL-Security Check Configuration, page 6 (optional)

Configuring the TTL-Security Check

To configure the BGP Support for TTL Security Check Feature, perform the steps in this section.

Prerequisites

- To maximize the effectiveness of this feature, we recommend that you configure it on each participating router. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router.

Restrictions

- The **neighbor ebgp-multihop** command is not needed when this feature is configured for a multihop peering session and should be disabled before configuring this feature.
- The effectiveness of this feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected peering sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of the local and remote network. This restriction also includes peers that are on the network segment between the local and remote network.

SUMMARY STEPS

1. **enable**
2. **trace** [**protocol**] *destination*
3. **configure terminal**
4. **router bgp** *as-number*
5. **neighbor** *ip-address* **ttl-security hops** *hop-count*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace** [**protocol**] *destination* **Example:** `Router# trace ip 10.1.1.1`	Discovers the routes of the specified protocol that packets will actually take when traveling to their destination. • Enter the **trace** command to determine the number of hops to the specified peer.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *as-number* **Example:** `Router(config)# router bgp 1`	Enters router configuration mode, and creates a BGP routing process.
Step 5	**neighbor** *ip-address* **ttl-security hops** *hop-count* **Example:** `Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2`	Configures the maximum number of hops that separate two peers. • The *hop-count* argument is set to number of hops that separate the local and remote peer. If the expected TTL value in the IP packet header is 254, then the number 1 should be configured for the *hop-count* argument. The range of values is a number from 1 to 254. • When this feature is enabled, BGP will accept incoming IP packets with a TTL value that is equal to or greater than the expected TTL value. Packets that are not accepted are silently discarded. • The example configuration sets the expected incoming TTL value to at least 253, which is 255 minus the TTL value of 2, and this is the minimum TTL value expected from the BGP peer. The local router will accept the peering session from the 10.1.1.1 neighbor only if it is 1 or 2 hops away.
Step 6	**end** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters privileged EXEC mode.

Examples

The following example sets the expected incoming TTL value for a directly connected eBGP peer. The *hop-count* argument is set to 2 configuring BGP to only accept IP packets with a TTL count in the header that is equal to or greater than 253. If the 10.1.1.1 neighbor is more than 2 hops away, the peering session will not be accepted.

```
neighbor 10.1.1.1 ttl-security hops 2
```

What to Do Next

The next task is to verify the TTL-security check configuration. Use the steps in the Verifying TTL-Security Check Configuration section.

Verifying the TTL-Security Check Configuration

You can verify the local configuration of this feature with the **show running-config** and **show ip bgp neighbors** commands.

SUMMARY STEPS

1. **enable**
2. **show running-config** [**interface** *type number*] [**linenum**] [**map-class**]
3. **show ip bgp neighbors** *neighbor-address* [**advertised-routes** | **dampened-routes** | **paths** *regular-expression* | **policy** | **received-routes** | **routes** | **received prefix-filter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [**interface type number**] [**linenum**] [**map-class**] **Example:** `Router# show running-config | begin bgp`	Displays the contents of the currently running configuration file. • The output of this command displays the configuration of the **neighbor ttl-security** command for each peer under the BGP configuration section. This section includes the neighbor address and the configured hop count.
Step 3	**show ip bgp neighbors** *neighbor-address* [**advertised-routes** | **dampened-routes** | **paths** {*regular-expression*} | **policy** | **received-routes** | **routes** | **received prefix-filter**] **Example:** `Router# show ip bgp neighbors 10.1.1.14`	Displays information about the TCP and BGP connections to neighbors. • The **show ip bgp neighbors** command displays "External BGP neighbor may be up to *number* hops away" when this feature is enabled. The *number* value represents the hop count. It is a number from 1 to 254.

Configuration Examples for the BGP Support for TTL Security Check Feature

The following examples show how to configure and verify this feature:

Configuring the TTL-Security Check: Example

The example configurations in this section show how to configure the BGP Support for TTL Security Check feature.

The following example uses the **trace** command to determine the hop count to an eBGP peer. The hop count number is displayed in the output for each networking device that IP packets traverse to reach the specified neighbor. In the example below, the hop count for the 10.1.1.1 neighbor is 1.

```
Router# trace ip 10.1.1.1

Type escape sequence to abort.
Tracing the route to 10.1.1.1

  1 10.1.1.1 0 msec *  0 msec
```

The following example sets the hop count to 2 for the 10.1.1.1 neighbor. Because the *hop-count* argument is set to 2, BGP will only accept IP packets with a TTL count in the header that is equal to or greater than 253.

```
Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
```

Verifying the TTL-Security Check Configuration: Example

The configuration of the BGP Support for TTL Security Check feature can be verified with the **show running-config** and **show ip bgp neighbors** commands. This feature is configured locally on each peer, so there is no remote configuration to verify.

The following is sample output from the **show running-config** command. The output shows that neighbor 10.1.1.1 is configured to establish or maintain the peering session only if the expected TTL count in the incoming IP packet is 253 or 254.

```
Router# show running-config | begin bgp

router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 ttl-security hops 2
 no auto-summary
.
.
.
```

The following is sample output from the **show ip bgp neighbors** command. The output shows that the local router will accept packets from the 10.1.1.1 neighbor if it is no more than 2 hops away. The configuration of this feature is displayed in the address family section of the output. The relevant line is bolded in the output.

```
Router# show ip bgp neighbors 10.1.1.1

BGP neighbor is 10.1.1.1,  remote AS 100, internal link
  BGP version 4, remote router ID 10.2.2.22
  BGP state = Established, up for 00:59:21
  Last read 00:00:21, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  2          2
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           226        227
    Route Refresh:          0          0
    Total:                228        229
  Default minimum time between advertisement runs is 5 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  Member of update-group 1
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              0          0
    Prefixes Total:                0          0
    Implicit Withdraw:             0          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          0
    Used as multipath:           n/a          0

                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    Total:                               0          0
  Number of NLRIs in the update sent: max 0, min 0

  Connections established 2; dropped 1
  Last reset 00:59:50, due to User reset
  External BGP neighbor may be up to 2 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.2.2.22, Local port: 179
Foreign host: 10.1.1.1, Foreign port: 11001

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0xCC28EC):
Timer          Starts    Wakeups            Next
Retrans            63          0             0x0
TimeWait            0          0             0x0
AckHold            62         50             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
```

```
iss:  712702676  snduna:  712703881  sndnxt:  712703881     sndwnd:  15180
irs: 2255946817  rcvnxt: 2255948041  rcvwnd:      15161  delrcvwnd:   1223

SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):
Rcvd: 76 (out of order: 0), with data: 63, total data bytes: 1223
Sent: 113 (retransmit: 0, fastretransmit: 0), with data: 62, total data bytes: 4
```

Additional References

The following sections provide references related to the BGP Support For TTL Security Check feature.

Related Documents

Related Topic	Document Title
BGP commands	• *Cisco IOS Release 12.0 Network Protocols Command Reference, Part 1* • *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols,* Release 12.2 • *Cisco IOS IP Command Reference, Volume 2 of 4: Routing Protocols,* Release 12.3T
BGP configuration tasks	• *Cisco IOS Release 12.0 Network Protocols Configuration Guide, Part 1* • *Cisco IOS IP Configuration Guide,* Release 12.2 • *Cisco IOS IP Configuration Guide,* Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 3682	The Generalized TTL Security Mechanism (GTSM)

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	TAC Home Page: http://www.cisco.com/public/support/tac/home.shtml BGP Support Page: http://www.cisco.com/cgi-bin/Support/browse/psp_view.pl?p=Internetworking:BGP

Command Reference

This section documents new and modified commands.

New Command

- **neighbor ttl-security**

Modified Command

- **show ip bgp neighbors**

neighbor ttl-security

To secure a Border Gateway Protocol (BGP) peering session and to configure the maximum number of hops that separate two external BGP (eBGP) peers, use the **neighbor ttl-security** command in address-family or router configuration mode. To disable this feature, use the **no** form of this command.

neighbor *neighbor-address* **ttl-security hops** *hop-count*

no neighbor *neighbor-address* **ttl-security hops** *hop-count*

Syntax Description

neighbor-address	IP address of the neighbor.
hops *hop-count*	Maxim number of hops that can separate the eBGP peer from the local router. The value for the *hop-count* argument is a number from 1 to 254.

Defaults

No default behavior or values

Command Modes

Address-family configuration
Router configuration

Command History

Release	Modification
12.0(27)S	This command was introduced.
12.3(7)T	This command was integrated into Cisco IOS Release 12.3(7)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(18)SXE	This command was integrated into Cisco IOS Release 12.2(18)SXE.

Usage Guidelines

The **neighbor ttl-security** command provides a lightweight security mechanism to protect BGP peering sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses in the packet headers.

This feature leverages designed behavior of IP packets by accepting only incoming IP packets with a TTL count that is equal to or greater than the locally configured value. Accurately forging the TTL count in an IP packet is generally considered to be impossible. Accurately forging a packet to match the TTL count from a trusted peer is not possible without internal access to the source or destination network.

This feature should be configured on each participating router. It secures the BGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the expected TTL value configured for the peering session. This feature only verifies the number of hops that separate the two BGP peers. The BGP peering session can still expire if keepalive packets are not received. If a packet is received with a TTL count that is less than expected number, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is not necessary.

To maximize the effectiveness of this feature, the *hop-count* value should be strictly configured to match the number of hops between the local and external network. However, you should also take path variation into account when configuring this feature for a multihop peering session.

The following restrictions apply to the configuration of this command:

- This feature is not supported for internal BGP (iBGP) peers or iBGP peer groups.
- The **neighbor ttl-security** command cannot be configured for a peer that is already configured with the **neighbor ebgp-multihop** command. The configuration of these commands is mutually exclusive, and only one of these commands is needed to enable a multihop eBGP peering session. An error message will be displayed in the console if you attempt to configure both commands for the same peering session.
- The effectiveness of this feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected peering sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of your network. This restriction also includes peers that are on the network segment between the source and destination network.

Examples

The following example sets the hop count to 2 for a directly connected neighbor. Because the *hop-count* argument is set to 2, BGP will accept only IP packets with a TTL count in the header that is equal to or greater than 253 (253 or 254). If a packet is received with any other TTL value in the IP packet header, the packet will be silently discarded.

```
neighbor 10.0.0.1 ttl-security hops 2
```

Related Commands

Command	Description
neighbor ebgp-multihop	Accepts or initiates BGP connections to external peers residing on networks that are not directly connected.
show ip bgp neighbors	Displays information about the TCP and BGP connections to neighbors.

show ip bgp neighbors

To display information about the TCP and Border Gateway Protocol (BGP) connections to neighbors, use the **show ip bgp neighbors** command in EXEC mode.

show ip bgp neighbors [*neighbor-address*] [**advertised-routes | dampened-routes | paths** *regexp* **| policy** [**detail**] **| received-routes | routes | received prefix-filter**]

Syntax Description

neighbor-address	(Optional) Address of the neighbor from which the local router has learned routes. If you omit this argument, all neighbors are displayed.
advertised-routes	(Optional) Displays all the routes the router has advertised to the neighbor.
dampened-routes	(Optional) Displays the dampened routes to the neighbor at the IP address specified.
paths *regexp*	(Optional) Regular expression that is used to match the paths received.
policy	(Optional) Displays neighbor policies per address family.
detail	(Optional) This keyword is used with the **policy** keyword to display more detailed policy information.
received-routes	(Optional) Displays all received routes (both accepted and rejected) from the specified neighbor.
routes	(Optional) Displays all routes that are received and accepted. The out that is generated by this keyword is a subset of the output from the **received-routes** keyword.
received prefix-filter	(Optional) Displays the configured prefix list filter for the specified IP address

Command Modes

EXEC

Command History

Release	Modification
10.0	This command was introduced.
11.2	The **received-routes** keyword was added.
12.2(4)T	The **received prefix-filter** keyword was added.
12.2(8)T	The **no-prepend** configuration option was added to the display output.
12.0(21)ST	This command was updated to display MPLS label information.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S. Support for the Cisco 12000 series routers (Engine 0 and Engine 2) was added. The **received prefix-filter** keyword was added.
12.0(22)S 12.2(15)T	Support for the BGP graceful restart capability was integrated into the output.
12.0(27)S 12.3(7)T 12.2(25)S	The **policy** and **detail** keywords were added.

Examples

The following is sample output from the **show ip bgp neighbors** command. This output shows that neighbor 10.1.1.1 is configured with the BGP Support for TTL Security Check feature. The expected TTL count is set to 2. If a packet is received from the 10.1.1.1 neighbor from over more than 2 hops, the packet will be silently discarded. The configuration of this feature is displayed in the address family section of the output. The relevant line is bolded in the output.

```
Router# show ip bgp neighbors 10.1.1.1

BGP neighbor is 10.1.1.1,  remote AS 100, internal link
  BGP version 4, remote router ID 10.2.2.22
  BGP state = Established, up for 00:59:21
  Last read 00:00:21, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  2          2
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           226        227
    Route Refresh:          0          0
    Total:                228        229
  Default minimum time between advertisement runs is 5 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  Member of update-group 1
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              0          0
    Prefixes Total:                0          0
    Implicit Withdraw:             0          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          0
    Used as multipath:           n/a          0

                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    Total:                               0          0
  Number of NLRIs in the update sent: max 0, min 0

  Connections established 2; dropped 1
  Last reset 00:59:50, due to User reset
  External BGP neighbor may be up to 2 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.2.2.22, Local port: 179
Foreign host: 10.1.1.1, Foreign port: 11001

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0xCC28EC):
Timer          Starts    Wakeups            Next
Retrans            63          0             0x0
TimeWait            0          0             0x0
AckHold            62         50             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
```

```
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss:  712702676  snduna:  712703881  sndnxt:  712703881     sndwnd:  15180
irs: 2255946817  rcvnxt: 2255948041  rcvwnd:      15161  delrcvwnd:   1223

SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):
Rcvd: 76 (out of order: 0), with data: 63, total data bytes: 1223
Sent: 113 (retransmit: 0, fastretransmit: 0), with data: 62, total data bytes: 4
```

Table 1 describes the significant fields shown in the display.

Table 1 show ip bgp neighbors Field Descriptions

Field	Description
BGP neighbor	IP address of the BGP neighbor and its autonomous system number. If the neighbor is in the same autonomous system as the router, then the link between them is internal; otherwise, it is considered external.
remote AS	Autonomous system of the neighbor.
external link	Indicates that this peer is an eBGP peer.
BGP version	BGP version being used to communicate with the remote router; the router ID (an IP address) of the neighbor is also specified.
remote router ID	IP address of the neighbor.
BGP state	Internal state of this BGP connection.
up for	Amount of time, in seconds, that the underlying TCP connection has been in existence.
Last read	Time that BGP last read a message from this neighbor.
hold time	Maximum amount of time that can elapse between messages from the peer.
keepalive interval	Time period, in seconds, between sending keepalive packets, which help ensure that the TCP connection is up.
Neighbor capabilities	BGP capabilities advertised and received from this neighbor.
MPLS Label capability	Indicates that MPLS labels are both sent and received by the eBGP peer.
Address family IPv4 Unicast:	IP Version 4 unicast-specific properties of this neighbor.
Graceful Restart Capability:	The status of the graceful restart capability. “Advertised and received” is displayed when the graceful restart capability has been successfully exchanged between 2 routers.
received	Number of total BGP messages received from this peer, including keepalives.
Notifications	Number of error messages received from the peer.
Sent	Total number of BGP messages that have been sent to this peer, including keepalives.
notifications	Number of error messages the router has sent to this peer.

Table 1 show ip bgp neighbors Field Descriptions (continued)

Field	Description
Route Refresh Request:	Number of route refresh requests sent and received from this neighbor.
advertisement runs	Value of the minimum advertisement interval (in seconds).
For address family:	Address family to which the following fields refer.
BGP table version	Indicates that the neighbor has been updated with this version of the primary BGP routing table.
neighbor version	Number used by the software to track the prefixes that have been sent and those that must be sent to this neighbor.
Community attribute	Appears if the **neighbor send-community** command is configured for this neighbor.
Inbound path policy	Indicates if an inbound policy is configured.
Outbound path policy	Indicates if an outbound policy is configured.
uni-in	Name of inbound route map for the unicast address family.
uni-out	Name of outbound route map for the unicast address family.
mul-in	Name of inbound route map for the multicast address family.
mul-out	Name of outbound route map for the multicast address family.
Sending Prefix & Label	Indicates that the eBGP peer sends MPLS labels with its routes.
accepted prefixes	Number of prefixes accepted.
Prefix advertised	Number of prefixes advertised.
suppressed	Number of prefixes suppressed.
withdrawn	Number of prefixes withdrawn.
Connections established	Number of times the router has established a TCP connection and the 2 peers have agreed to speak BGP with each other.
dropped	Number of times that a good connection has failed or been taken down.
Last reset	Elapsed time (in hh:mm:ss) since this peering session was last reset.
External BGP neighbor may be...	Indicates that the BGP TTL security check is enabled. The maximum number hops that can separate the local and remote peer is displayed on this line.
Connection state	State of BGP peer.
unread input bytes	Number of bytes of packets still to be processed.
Local host, Local port	Peering address of local router, plus port.
Foreign host, Foreign port	Peering address of the neighbor.
Event Timers	Table displays the number of starts and wakeups for each timer.
iss	Initial send sequence number.
snduna	Last send sequence number the local host sent but has not received an acknowledgment for.
sndnxt	Sequence number the local host will send next.
sndwnd	TCP window size of the remote host.

Table 1 show ip bgp neighbors Field Descriptions (continued)

Field	Description
irs	Initial receive sequence number.
rcvnxt	Last receive sequence number the local host has acknowledged.
rcvwnd	TCP window size of the local host.
delrcvwnd	Delayed receive window—data the local host has read from the connection but has not yet subtracted from the receive window the host has advertised to the remote host. The value in this field gradually increases until it is larger than a full-sized packet, at which point it is applied to the rcvwnd field.
SRTT	A calculated, smoothed round-trip timeout.
RTTO	Round-trip timeout.
RTV	Variance of the round-trip time.
KRTT	New round-trip timeout (using the Karn algorithm). This field separately tracks the round-trip time of packets that have been re-sent.
minRTT	Smallest recorded round-trip timeout (hard wire value used for calculation).
maxRTT	Largest recorded round-trip timeout.
ACK hold	Time the local host will delay an acknowledgment in order to “piggyback” data on it.
Flags	IP precedence of the BGP packets.
Datagrams: Rcvd	Number of update packets received from a neighbor.
with data	Number of update packets received with data.
total data bytes	Total bytes of data.
Sent	Number of update packets sent.
with data	Number of update packets with data sent.
total data bytes	Total number of data bytes.