

Network Configuration Example

Configuring Security Options for BGP with TCP



Published: 2014-01-10

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Network Configuration Example Configuring Security Options for BGP with TCP
NCE0047
Copyright © 2014, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction	1
Understanding Security Options for BGP with TCP	1
Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers	1
Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List	7
Example: Limiting TCP Segment Size for BGP	10

Introduction

This document provides an overview and describes three methods for configuring security for routers using BGP with TCP. The first method shows how to create a firewall filter that blocks TCP access to a port for all requesters other than specified peers. The second method shows how to configure a firewall filter that limits certain TCP and Internet Control Message Protocol (ICMP) traffic based on a prefix list. The third method shows how to configure a router to avoid ICMP vulnerability issues by limiting TCP segment size when using maximum transmission unit (MTU) discovery.

Understanding Security Options for BGP with TCP

Among routing protocols, BGP is unique in using TCP as its transport protocol. BGP peers are established by manual configuration between routing devices to create a TCP session on port 179. A BGP-enabled device periodically sends keepalive messages to maintain the connection.

Over time, BGP has become the dominant interdomain routing protocol on the Internet. However, it has limited guarantees of stability and security. Configuring security options for BGP must balance suitable security measures with acceptable costs. No one method has emerged as superior to other methods. Each network administrator must configure security measures that meet the needs of the network being used.

For detailed information about the security issues associated with BGP's use of TCP as a transport protocol, see RFC 4272, *BGP Security Vulnerabilities Analysis*.

Related Documentation

- [Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers on page 1](#)
- [Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 7](#)
- [Example: Limiting TCP Segment Size for BGP on page 10](#)

Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers

This example shows how to configure a standard stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except from specified BGP peers.

- [Requirements on page 2](#)
- [Overview on page 2](#)
- [Configuration on page 2](#)
- [Verification on page 5](#)

Requirements

No special configuration beyond device initialization is required before you configure this example.

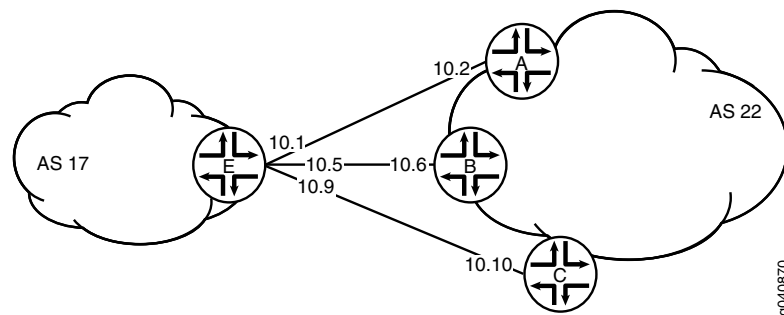
Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except the specified BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the directly connected interfaces on Device A and Device B to the destination port number 179.

Figure 1 on page 2 shows the topology used in this example. Device C attempts to make a TCP connection to Device E. Device E blocks the connection attempt. This example shows the configuration on Device E.

Figure 1: Typical Network with BGP Peer Sessions



Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device C

```
set interfaces ge-1/2/0 unit 10 description to-E
set interfaces ge-1/2/0 unit 10 family inet address 10.10.10.10/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 17
set protocols bgp group external-peers neighbor 10.10.10.9
set routing-options autonomous-system 22
```

Device E

```
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-1/2/1 unit 5 description to-B
set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-1/0/0 unit 9 description to-C
set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 2 family inet filter input filter_bgp179
set interfaces lo0 unit 2 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
```

```
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set routing-options autonomous-system 17
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.2/32
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.6/32
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then accept
set firewall family inet filter filter_bgp179 term 2 then reject
```

Configuring Device E

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device E with a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requestors except specified BGP peers:

1. Configure the interfaces.

```
[edit]
user@E# set interfaces ge-1/2/0 unit 0 description to-A
user@E# set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30

user@E# set interfaces ge-1/2/1 unit 5 description to-B
user@E# set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30

user@E# set interfaces ge-1/0/0 unit 9 description to-C
user@E# set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
```

2. Configure BGP.

```
[edit protocols bgp group external-peers]
user@E# set type external
user@E# set peer-as 22
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10
```

3. Configure the autonomous system number.

```
[edit routing-options]
user@E# set autonomous-system 17
```

4. Define the filter term that accepts TCP connection attempts to port 179 from the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@E# set term 1 from source-address 10.10.10.2/32
user@E# set term 1 from source-address 10.10.10.6/32
user@E# set term 1 from destination-port bgp
user@E# set term 1 then accept
```

5. Define the other filter term to reject packets from other sources.

```
[edit firewall family inet filter filter_bgp179]
```

```
user@E# set term 2 then reject
```

6. Apply the firewall filter to the loopback interface.

```
[edit interfaces lo0 unit 2 family inet]
user@E# set filter input filter_bgp179
user@E# set address 192.168.0.1/32
```

Results From configuration mode, confirm your configuration by entering the **show firewall**, **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@E# show firewall
family inet {
  filter filter_bgp179 {
    term 1 {
      from {
        source-address {
          10.10.10.2/32;
          10.10.10.6/32;
        }
        destination-port bgp;
      }
      then accept;
    }
    term 2 {
      then {
        reject;
      }
    }
  }
}
```

```
user@E# show interfaces
lo0 {
  unit 2 {
    family inet {
      filter {
        input filter_bgp179;
      }
      address 192.168.0.1/32;
    }
  }
}
ge-1/2/0 {
  unit 0 {
    description to-A;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
ge-1/2/1 {
  unit 5 {
    description to-B;
```

```
        family inet {
            address 10.10.10.5/30;
        }
    }
}
ge-1/0/0 {
    unit 9 {
        description to-C;
        family inet {
            address 10.10.10.9/30;
        }
    }
}
```

```
user@E# show protocols
```

```
bgp {
    group external-peers {
        type external;
        peer-as 22;
        neighbor 10.10.10.2;
        neighbor 10.10.10.6;
        neighbor 10.10.10.10;
    }
}
```

```
user@E# show routing-options
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the Filter Is Configured on page 5](#)
- [Verifying the TCP Connections on page 5](#)
- [Monitoring Traffic on the Interfaces on page 6](#)

Verifying That the Filter Is Configured

Purpose Make sure that the filter is listed in output of the **show firewall filter** command.

Action user@E> show firewall filter filter_bgp179
Filter: filter_bgp179

Verifying the TCP Connections

Purpose Verify the TCP connections.

Action From operational mode, run the **show system connections extensive** command on Device C and Device E.

The output on Device C shows the attempt to establish a TCP connection. The output on Device E shows that connections are established with Device A and Device B only.

```
user@C> show system connections extensive | match 10.10.10
```

```
tcp4      0      0 10.10.10.9.51872    10.10.10.10.179    SYN_SENT
```

```
user@E> show system connections extensive | match 10.10.10
```

```
tcp4      0      0 10.10.10.5.179      10.10.10.6.62096    ESTABLISHED
tcp4      0      0 10.10.10.6.62096    10.10.10.5.179      ESTABLISHED
tcp4      0      0 10.10.10.1.179      10.10.10.2.61506    ESTABLISHED
tcp4      0      0 10.10.10.2.61506    10.10.10.1.179      ESTABLISHED
```

Monitoring Traffic on the Interfaces

Purpose Use the **monitor traffic** command to compare the traffic on an interface that establishes a TCP connection with the traffic on an interface that does not establish a TCP connection.

Action From operational mode, run the **monitor traffic** command on the Device E interface to Device B and on the Device E interface to Device C. The following sample output verifies that in the first example, acknowledgment (**ack**) messages are received. In the second example, **ack** messages are not received.

```
user@E> monitor traffic size 1500 interface ge-1/2/1.5
```

```
19:02:49.700912 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P
3330573561:3330573580(19) ack 915601686 win 16384 <nop,nop,timestamp 1869518816
1869504850>: BGP, length: 19
19:02:49.801244 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 19 win 16384
<nop,nop,timestamp 1869518916 1869518816>
19:03:03.323018 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: P 1:20(19) ack 19 win
16384 <nop,nop,timestamp 1869532439 1869518816>: BGP, length: 19
19:03:03.422418 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: . ack 20 win 16384
<nop,nop,timestamp 1869532539 1869532439>
19:03:17.220162 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P 19:38(19) ack 20 win
16384 <nop,nop,timestamp 1869546338 1869532439>: BGP, length: 19
19:03:17.320501 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 38 win 16384
<nop,nop,timestamp 1869546438 1869546338>
```

```
user@E> monitor traffic size 1500 interface ge-1/0/0.9
```

```
18:54:20.175471 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869009240 0,sackOK,eol>
18:54:23.174422 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869012240 0,sackOK,eol>
18:54:26.374118 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869015440 0,sackOK,eol>
18:54:29.573799 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:32.773493 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:35.973185 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
```

Related Documentation

- [Understanding Security Options for BGP with TCP on page 1](#)
- [Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 7](#)

- [Example: Limiting TCP Segment Size for BGP on page 10](#)

Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List

This example shows how to configure a standard stateless firewall filter that limits certain TCP and Internet Control Message Protocol (ICMP) traffic destined for the Routing Engine by specifying a list of prefix sources that contain allowed BGP peers.

- [Requirements on page 7](#)
- [Overview on page 7](#)
- [Configuration on page 7](#)
- [Verification on page 9](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except BGP peers that have a specified prefix.

A source prefix list, **plist_bgp179**, is created that specifies the list of source prefixes that contain allowed BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the source prefix list **plist_bgp179** to the destination port number 179.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

- [Configure the Filter on page 8](#)
- [Results on page 8](#)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options prefix-list plist_bgp179 apply-path "protocols bgp group <*> neighbor <*>"
set firewall family inet filter filter_bgp179 term 1 from source-address 0.0.0.0/0
set firewall family inet filter filter_bgp179 term 1 from source-prefix-list plist_bgp179 except
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then reject
set firewall family inet filter filter_bgp179 term 2 then accept
set interfaces lo0 unit 0 family inet filter input filter_bgp179
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Configure the Filter

Step-by-Step Procedure The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the filter:

1. Expand the prefix list **bgp179** to include all prefixes pointed to by the BGP peer group defined by **protocols bgp group <*> neighbor <*>**.

```
[edit policy-options prefix-list plist_bgp179]
user@host# set apply-path "protocols bgp group <*> neighbor <*>"
```

2. Define the filter term that rejects TCP connection attempts to port 179 from all requesters except the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term1 from source-address 0.0.0.0/0
user@host# set term term1 from source-prefix-list bgp179 except
user@host# set term term1 from destination-port bgp
user@host# set term term1 then reject
```

3. Define the other filter term to accept all packets.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term2 then accept
```

4. Apply the firewall filter to the loopback interface.

```
[edit interfaces lo0 unit 0 family inet]
user@host# set filter input filter_bgp179
user@host# set address 127.0.0.1/32
```

Results

From configuration mode, confirm your configuration by entering the **show firewall**, **show interfaces**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter filter_bgp179 {
    term 1 {
      from {
        source-address {
          0.0.0.0/0;
        }
        source-prefix-list {
          plist_bgp179 except;
        }
        destination-port bgp;
      }
      then {
        reject;
      }
    }
  }
}
```

```

    }
    term 2 {
        then {
            accept;
        }
    }
}

user@host# show interfaces
lo0 {
    unit 0 {
        family inet {
            filter {
                input filter_bgp179;
            }
            address 127.0.0.1/32;
        }
    }
}

user@host# show policy-options
prefix-list plist_bgp179 {
    apply-path "protocols bgp group <*> neighbor <*>";
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Repeat the procedure, where appropriate, for every BGP-enabled device in the network, using the appropriate interface names and addresses for each BGP-enabled device.

Verification

Confirm that the configuration is working properly.

Displaying the Firewall Filter Applied to the Loopback Interface

Purpose Verify that the firewall filter **filter_bgp179** is applied to the IPv4 input traffic at logical interface **lo0.0**.

Action Use the **show interfaces statistics** operational mode command for logical interface **lo0.0**, and include the **detail** option. Under the **Protocol inet** section of the command output section, the **Input Filters** field displays the name of the stateless firewall filter applied to the logical interface in the input direction:

```

[edit]
user@host> show interfaces statistics lo0.0 detail
Logical interface lo0.0 (Index 321) (SNMP ifIndex 16) (Generation 130)
  Flags: SNMP-Traps Encapsulation: Unspecified
  Traffic statistics:
    Input bytes : 0
    Output bytes : 0
    Input packets: 0
    Output packets: 0
  Local statistics:
    Input bytes : 0
    Output bytes : 0

```

```
Input packets:          0
Output packets:         0
Transit statistics:
Input bytes :           0          0 bps
Output bytes :          0          0 bps
Input packets:         0          0 pps
Output packets:        0          0 pps
Protocol inet, MTU: Unlimited, Generation: 145, Route table: 0
  Flags: Sendbroadcast-pkt-to-re
  Input Filters: filter_bgp179
  Addresses, Flags: Primary
    Destination: Unspecified, Local: 127.0.0.1, Broadcast: Unspecified,
    Generation: 138
```

**Related
Documentation**

- [Understanding Security Options for BGP with TCP on page 1](#)
- [Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers on page 1](#)
- [Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 7](#)

Example: Limiting TCP Segment Size for BGP

This example shows how to avoid Internet Control Message Protocol (ICMP) vulnerability issues by limiting TCP segment size when you are using maximum transmission unit (MTU) discovery. Using MTU discovery on TCP paths is one method of avoiding BGP packet fragmentation.

- [Requirements on page 10](#)
- [Overview on page 10](#)
- [Configuration on page 11](#)
- [Verification on page 13](#)
- [Troubleshooting on page 13](#)

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

TCP negotiates a maximum segment size (MSS) value during session connection establishment between two peers. The MSS value negotiated is primarily based on the maximum transmission unit (MTU) of the interfaces to which the communicating peers are directly connected. However, due to variations in link MTU on the path taken by the TCP packets, some packets in the network that are well within the MSS value might be fragmented when the packet size exceeds the link's MTU.

To configure the TCP MSS value, include the **tcp-mss** statement with a segment size from 1 through 4096.

If the router receives a TCP packet with the SYN bit and the MSS option set, and the MSS option specified in the packet is larger than the MSS value specified by the **tcp-mss** statement, the router replaces the MSS value in the packet with the lower value specified by the **tcp-mss** statement.

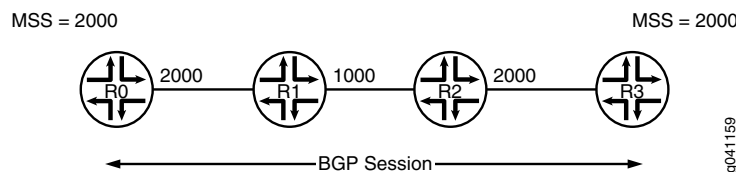
The configured MSS value is used as the maximum segment size for the sender. The assumption is that the TCP MSS value used by the sender to communicate with the BGP neighbor is the same as the TCP MSS value that the sender can accept from the BGP neighbor. If the MSS value from the BGP neighbor is less than the MSS value configured, the MSS value from the BGP neighbor is used as the maximum segment size for the sender.

This feature is supported with TCP over IPv4 and TCP over IPv6.

Topology Diagram

Figure 2 on page 11 shows the topology used in this example.

Figure 2: TCP Maximum Segment Size for BGP



Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
R0
set interfaces fe-1/2/0 unit 1 family inet address 1.1.0.1/30
set interfaces lo0 unit 1 family inet address 10.255.14.179/32
set protocols bgp group-int tcp-mss 2020
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.179
set protocols bgp group int mtu-discovery
set protocols bgp group int neighbor 10.255.71.24 tcp-mss 2000
set protocols bgp group int neighbor 10.255.14.177
set protocols bgp group int neighbor 10.0.14.4 tcp-mss 4000
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface 10.255.14.179
set routing-options autonomous-system 65000
```

Step-by-Step Procedure The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Router R0:

1. Configure the interfaces.

[edit interfaces]

```
user@R0# set fe-1/2/0 unit 1 family inet address 1.1.0.1/30
user@R0# set lo0 unit 1 family inet address 10.255.14.179/32
```

2. Configure an interior gateway protocol (IGP), OSPF in this example.

```
[edit protocols ospf area 0.0.0.0]
user@R0# set interface fe-1/2/0.1
user@R0# set interface 10.255.14.179
```

3. Configure one or more BGP groups.

```
[edit protocols bgp group int]
user@R0# set type internal
user@R0# set local-address 10.255.14.179
```

4. Configure MTU discovery to prevent packet fragmentation.

```
[edit protocols bgp group int]
user@R0# set mtu-discovery
```

5. Configure the BGP neighbors, with the TCP MSS set globally for the group or specifically for the various neighbors.

```
[edit protocols bgo group int]
user@R0# set tcp-mss 2020
user@R0# set neighbor 10.255.14.177
user@R0# set neighbor 10.255.71.24 tcp-mss 2000
user@R0# set neighbor 10.0.14.4 tcp-mss 4000
```



NOTE: The TCP MSS neighbor setting overrides the group setting.

6. Configure the local autonomous system.

```
[edit routing-options]
user@R0# set autonomous-system 65000
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 1.1.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.255.14.179/32;
    }
  }
}
```



```
user@R0# show protocols
bgp {
  group int {
    type internal;
    local-address 10.255.14.179;
    mtu-discovery;
    tcp-mss 2020;
    neighbor 10.255.71.24 {
      tcp-mss 2000;
    }
    neighbor 10.255.14.177;
    neighbor 10.0.14.4 {
      tcp-mss 4000;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.1;
    interface 10.255.14.179;
  }
}

user@R0# show routing-options
autonomous-system 65000;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, run the following commands:

- **show system connections extensive | find <neighbor-address>**, to check the negotiated TCP MSS value.
- **monitor traffic interface**, to monitor BGP traffic and to make sure that the configured TCP MSS value is used as the MSS option in the TCP SYN packet.

Troubleshooting

- [MSS Calculation with MTU Discovery on page 13](#)

MSS Calculation with MTU Discovery

Problem Consider an example in which two routing devices (R1 and R2) have an internal BGP (IBGP) connection. On both of the routers, the connected interfaces have 4034 as the IPv4 MTU.

```
user@R1# show protocols bgp | display set
[edit]
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 45.45.45.2
set protocols bgp group ibgp mtu-discovery
set protocols bgp group ibgp neighbor 45.45.45.1

user@R1# run show interfaces xe-0/0/3 extensive | match mtu
```

```

Link-level type: Ethernet, MTU: 4048, LAN-PHY mode, Speed: 10Gbps,
FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Protocol inet, MTU: 4034, Generation: 180, Route table: 0
Protocol multiservice, MTU: Unlimited, Generation: 181, Route table: 0

```

In the following packet capture on Device R1, the negotiated MSS is 3994. In the **show system connections extensive** information for MSS, it is set to 2048.

```

05:50:01.575218 Out
  Juniper PCAP Flags [Ext], PCAP Extension(s) total length 16
    Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
    Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
    Device Interface Index Extension TLV #1, length 2, value: 137
    Logical Interface Index Extension TLV #4, length 4, value: 69
  -----original packet-----
  00:21:59:e1:e8:03 > 00:19:e2:20:79:01, ethertype IPv4 (0x0800), length
78: (tos 0xc0, ttl 64, id 53193, offset 0, flags [DF], proto: TCP (6), length:
64) 45.45.45.2.62840 > 45.45.45.1.bgp: S 2939345813:2939345813(0) win 16384 **mss
3994,nop,wscale 0,nop,nop,timestamp 70559970 0,sackOK,eol>
05:50:01.575875 In
  Juniper PCAP Flags [Ext, no-L2, In], PCAP Extension(s) total length 16
    Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
    Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
    Device Interface Index Extension TLV #1, length 2, value: 137
    Logical Interface Index Extension TLV #4, length 4, value: 69
  -----original packet-----
  PFE proto 2 (ipv4): (tos 0xc0, ttl 255, id 37709, offset 0, flags [DF], proto:
TCP (6), length: 64) 45.45.45.1.bgp > 45.45.45.2.62840: S 2634967984:2634967984(0)
ack 2939345814 win 16384 **mss 3994,nop,wscale 0,nop,nop,timestamp 174167273
70559970,sackOK,eol>

```

user@R1# run show system connections extensive | find 45.45

```

tcp4      0      0 45.45.45.2.62840          45.45.45.1.179
ESTABLISHED
  sndsbcc:      0 sndsbmbcnt:      0 sndsbmbmax:    131072
sndsblwat:    2048 sndsbhiwat:    16384
rcvsbcc:      0 rcvsbmbcnt:      0 rcvsbmbmax:    131072
rcvsblwat:      1 rcvsbhiwat:    16384
  proc id:    19725  proc name:      rpd
    iss: 2939345813  sndup: 2939345972
  snduna: 2939345991  sndnxt: 2939345991  sndwnd:    16384
  sndmax: 2939345991  sndcwnd:    10240  sndssthresh: 1073725440
    irs: 2634967984  rcvup: 2634968162
  rcvnxt: 2634968162  rcvadv: 2634984546  rcvwnd:    16384
    rtt:      0      srtt:    1538      rttv:    1040
  rxtcur:    1200  rxtshift:      0      rtseq: 2939345972
  rttmin:    1000  mss:      2048

```

Solution This is expected behavior with Junos OS. The MSS value is equal to the MTU value minus the IP or IPv6 and TCP headers. This means that the MSS value is generally 40 bytes less than the MTU (for IPv4) and 60 bytes less than the MTU (for IPv6). This value is negotiated between the peers. In this example, it is $4034 - 40 = 3994$. Junos OS then rounds this value to a multiple of 2 KB. The value is $3994 / 2048 * 2048 = 2048$. So it is not necessary to see same MSS value with in the **show system connections** output.

$3994 / 2048 = 1.95$

1.95 is rounded to 1.

$1 * 2048 = 2048$

**Related
Documentation**

- [Understanding Security Options for BGP with TCP on page 1](#)
- [Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers on page 1](#)
- [Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 7](#)

