Engineering
Simplicity

Network Configuration Example

Configuring Branch SRX Series for MPLS over GRE with IPsec Segmentation

Published
2023-09-26

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

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

Table of Contents

About This Guide

This network configuration example provides an overview of simplified MPLS over IPsec over 1500-byte media. It also contains a sample use case showing how to provide simplified configuration for VPLS or Layer 3 VPN services with GRE through IPsec tunneling, over 1500-byte media (Internet).

1

CHAPTER

Configuring Branch SRX Series for MPLS over GRE with IPsec Segmentation

About This Network Configuration Example

This network configuration example provides an overview of simplified MPLS over IPsec over 1500-byte media. It also contains a sample use case showing how to provide simplified configuration for VPLS or Layer 3 VPN services with GRE through IPsec tunneling, over 1500-byte media (Internet).

This document complements the configuration guidance provided in Example: Configuring Selective Packet Services and further explains the MPLS through IPsec over 1500byte media fragmentation and reassembly use case scenario.

Use Case for MPLS Through IPsec over 1500-byte Media

Use selective packet services in a single routing instance (the default one) without utilizing lt interfaces. You can perform IPsec encapsulated packet fragmentation on the outgoing physical interface of the sending device and reassembly on the receiving device before the IPsec decryption.

RELATED DOCUMENTATION

Simplified MPLS Through IPsec over 1500-byte Media Overview | 2
Example: Configuring MPLS over GRE with IPsec Fragmentation and Reassembly | 3

Simplified MPLS Through IPsec over 1500-byte Media Overview

RELATED DOCUMENTATION

Example: Configuring MPLS over GRE with IPsec Fragmentation and Reassembly | 3

Example: Configuring MPLS over GRE with IPsec Fragmentation and Reassembly

IN THIS SECTION

This example is based on a need to support a standard 1,500 byte MTU to virtual private network (VPN) clients that are supported by GRE over IPsec tunnels, when the WAN provider does not offer a Jumbo MTU option. The traffic forwarded over the 1500-byte WAN link can be dropped because the protocol encapsulation overhead (Layer 2, MPLS, GRE and IPsec) results in a frame that exceeds the WAN link MTU.

MTU related drops are mostly an issue for traffic that cannot be fragmented. For example, IP traffic that is marked as do-not-fragment, or Layer 2 VPN/VPLS traffic, which by its nature, cannot be fragmented. For performance reasons, many IPsec configurations block post encryption fragmentation, resulting in packet drop.

This document provide a solution to this problem by showing you how to configure an IPsec tunnel to perform post-fragmentation on traffic that is otherwise not able to be fragmented. In this case you trade encryption performance by forcing post-fragmentation against having to reduce the MTU of your VPN clients to prevent MTU related drops.

This example shows how to configure selective packet services mode using a single routing instance (the default one) to process VPN traffic into packet mode. In packet mode security zones are bypassed. This means that the Layer 2 and Layer 3 VRF interfaces are not placed into a security zone and no policy is needed to allow them to communicate through the internet zone.

Using the steps in this example you can perform IPsec encapsulated packet fragmentation on the outgoing physical interface of the sending device and reassembly on the receiving device before IPsec decryption.

NOTE: The reassembly of fragmented packets uses a lot of device resources, and the performance of the device will be slower than with nonfragmented traffic. When possible you should configure a jumbo MTU on the WAN interface to avoid the need for fragmentation. This example shows you how to provide a standard 1,500 byte MTU to client devices that block fragmentation when using IPsec over a WAN connection that does not offer jumbo support.

The topic includes the following sections:

Requirements

This example uses the following hardware and software components:

- Two SRX Series Services Gateways
- Junos OS Release 11.4 or later
  - This example has been revalidated on Junos OS Release 20.3R1

NOTE: For this example to work as documented you must ensure that your SRX configuration does not have any interfaces with `family ethernet-switching` enabled. Using `family ethernet-switching` puts the SRX device into mixed mode operation. This example is based on the route mode of operation. For details on route and mixed modes of operation see Understanding Layer 2 Interfaces on Security Devices. In addition, we tested this example with the factory default settings for the `edit protocols l2-learning` hierarchy.

Overview and Topology

IN THIS SECTION

This example includes the following configurations:

- Configure interfaces for the appropriate protocol encapsulation and maximum transmission unit (MTU) value.

- Apply firewall filter on the ge-0/0/0.10 interface to set the packet mode. Configure the WAN facing interface ge-0/0/1.0 with a 1,524 byte MTU.

- Set a large MTU value to GRE and IPsec logical interfaces to avoid IPsec fragmentation at logical interfaces. The GRE encapsulated traffic is tunneled inside IPsec.

- Add the MPLS family to the GRE interface gr-0/0/0, and apply firewall filters to enable packet mode.

- Configure an IPsec tunnel on the device with the `df-bit clear` option in the IPsec VPN configuration to allow fragmentation of oversized IPsec packets on the outgoing ge-0/0/1.0 interface. This setting allows the SRX device to perform fragmentation post IPsec encryption for VPN client traffic that is marked with the do not fragment (DNF) bit. VPN client traffic that is not marked as DNF is fragmented prior to IPsec encryption to improve performance.

- Configure all noncustomer- facing interfaces such as ge-0/0/1.0, gr-0/0/0.0, lo0.0, and st0.0 in a single security zone called “Internet”. A single security zone is used in this example to keep the focus on fragmentation issues with MPLS over GRE over IPSec. Security can be enhanced by placing the device into flow-mode for MPLS, and then placing the customer-facing interfaces into a zone. Once in a zone, security policies can control communications, and evoke advanced features like IDP and application recognition. For more information see Security Zones.

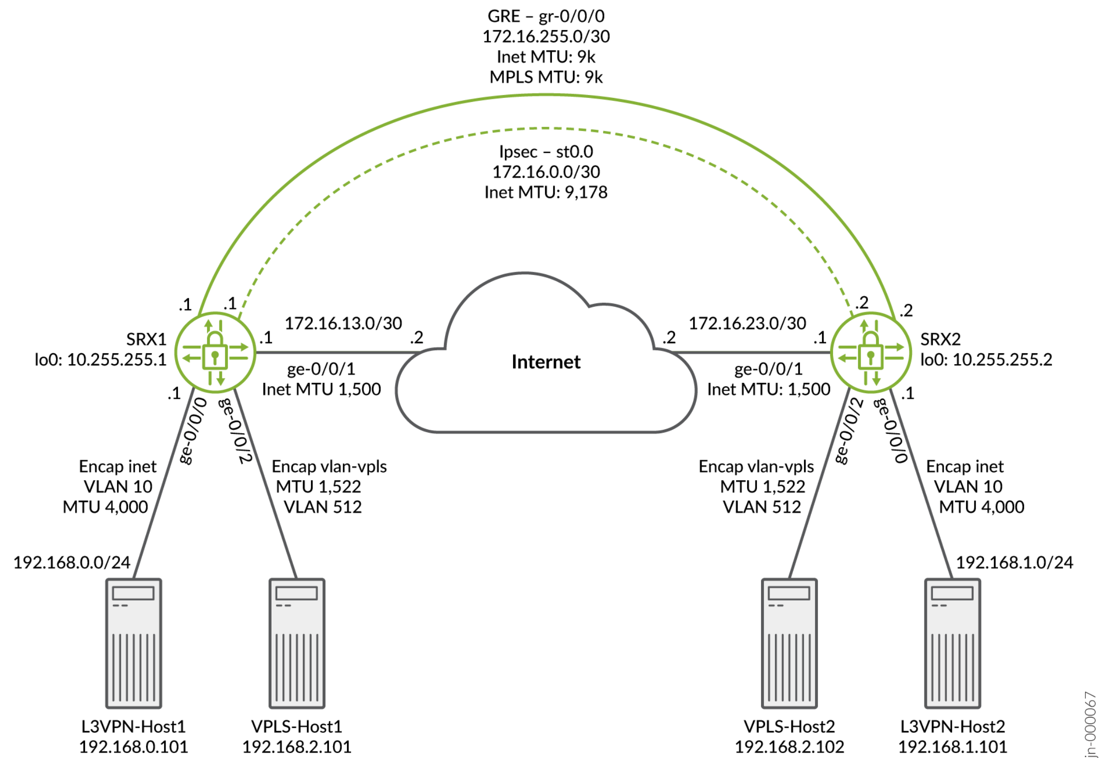
- Configure a policy to permit all (intrazone) traffic.

- Configure OSPF for lo0.0 address distribution, LDP for label distribution/MPLS transport, and IBGP with the `inet-vpn` and `l2vpn` families to support the VPN clients.

- Configure two routing instances, one for a Layer 3 VPN and one for a Layer 2 VPLS service.

Figure 1 on page 6 shows the topology for this example.

Figure 1: MPLS Over GRE Over IPsec Tunnels Example Topology



This example focuses on VPLS and a Layer 3 VPN over an IPsec tunnel. Layer 2 Circuits are also supported. For a Layer 2 circuit you need to configure both a family MPLS filter and a family CCC filter. The filters are used to evoke packet mode processing in order to support fragmentation over IPsec.

Topology

Table 1 on page 7 provides a summary of the parameters used in this topology for the PE1 device. You can adapt the parameters for the PE2 device, or use the PE2 quick configuration provided below.

Table 1: Components of the Topology

Components	Description
PE1	PE1 SRX Series Firewall: ge-0/0/0.10: • IP address: 192.168.0.1/24 • Customer-facing L3VPN interface • `input packet-mode-inet`: inet family in packet mode • MTU: 4k
	ge-0/0/2.11: • Customer-facing VPLS interface • `vlan-vpls`: VPLS encapsulation • MTU: 1,522
	ge-0/0/1.0: • Outgoing interface • IP address: 172.16.13.1/30 • MTU: 1,514
	gr-0/0/0: • Core interface connecting to MPLS • IP address: 172.16.255.1/30 • `input packet-mode`: MPLS family in packet mode • Inet MTU: 9k

Table 1: Components of the Topology *(Continued)*

Components	Description
	lo0: • Logical Interface • IP address: 10.255.255.1/32
	st0.0: • Tunnel interface • IP address: 172.16.0.1/30 • Inet MTU: 9,178
	• `df-bit clear` — This option clears the do not fragment (DF) bit in the outgoing packet header • `L3VPN`— Routing instance for Layer3 VPN application • `VPLS`— Routing instance for VPLS application

Configuration

IN THIS SECTION

Procedure

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, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

The configuration for the SRX1 (PE1) device:

```
set system host-name SRX1
set security ike policy standard mode main
set security ike policy standard proposal-set standard
set security ike policy standard pre-shared-key ascii-text "$9$1OsIclKvL7NblegoGUHk"
set security ike gateway srx-2 ike-policy standard
set security ike gateway srx-2 address 172.16.23.1
set security ike gateway srx-2 external-interface ge-0/0/1.0
set security ipsec policy standard proposal-set standard
set security ipsec vpn ipsec-vpn-1 bind-interface st0.0
set security ipsec vpn ipsec-vpn-1 df-bit clear
set security ipsec vpn ipsec-vpn-1 ike gateway srx-2
set security ipsec vpn ipsec-vpn-1 ike ipsec-policy standard
set security ipsec vpn ipsec-vpn-1 establish-tunnels immediately
set security policies from-zone Internet to-zone Internet policy Internet match source-address
any
set security policies from-zone Internet to-zone Internet policy Internet match destination-
address any
set security policies from-zone Internet to-zone Internet policy Internet match application any
set security policies from-zone Internet to-zone Internet policy Internet then permit
set security zones security-zone Internet host-inbound-traffic system-services all
set security zones security-zone Internet host-inbound-traffic protocols all
set security zones security-zone Internet interfaces ge-0/0/1.0
set security zones security-zone Internet interfaces gr-0/0/0.0
set security zones security-zone Internet interfaces lo0.0
set security zones security-zone Internet interfaces st0.0
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 mtu 4000
set interfaces ge-0/0/0 description L3VPN
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet filter input packet-mode-inet
set interfaces ge-0/0/0 unit 10 family inet address 192.168.0.1/24
set interfaces gr-0/0/0 description "MPLS core facing interface"
```

```
set interfaces gr-0/0/0 unit 0 tunnel source 172.16.0.1
set interfaces gr-0/0/0 unit 0 tunnel destination 172.16.0.2
set interfaces gr-0/0/0 unit 0 family inet mtu 9000
set interfaces gr-0/0/0 unit 0 family inet address 172.16.255.1/30
set interfaces gr-0/0/0 unit 0 family mpls mtu 9000
set interfaces gr-0/0/0 unit 0 family mpls filter input packet-mode
set interfaces ge-0/0/1 description Internet
set interfaces ge-0/0/1 mtu 1514
set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/30
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 mtu 1522
set interfaces ge-0/0/2 encapsulation vlan-vpls
set interfaces ge-0/0/2 unit 11 description VPLS
set interfaces ge-0/0/2 unit 11 encapsulation vlan-vpls
set interfaces ge-0/0/2 unit 11 vlan-id 512
set interfaces lo0 unit 0 family inet address 10.255.255.1/32
set interfaces st0 unit 0 family inet mtu 9178
set interfaces st0 unit 0 family inet address 172.16.0.1/30
set firewall family inet filter packet-mode-inet term all-traffic then packet-mode
set firewall family inet filter packet-mode-inet term all-traffic then accept
set firewall family mpls filter packet-mode term all-traffic then packet-mode
set firewall family mpls filter packet-mode term all-traffic then accept
set routing-instances L3VPN routing-options auto-export
set routing-instances L3VPN interface ge-0/0/0.10
set routing-instances L3VPN instance-type vrf
set routing-instances L3VPN route-distinguisher 10.255.255.1:1000
set routing-instances L3VPN vrf-target target:65100:1000
set routing-instances L3VPN vrf-table-label
set routing-instances VPLS protocols vpls site 1 interface ge-0/0/2.11
set routing-instances VPLS protocols vpls site 1 site-identifier 1
set routing-instances VPLS protocols vpls no-tunnel-services
set routing-instances VPLS protocols vpls mac-tlv-receive
set routing-instances VPLS protocols vpls mac-tlv-send
set routing-instances VPLS interface ge-0/0/2.11
set routing-instances VPLS instance-type vpls
set routing-instances VPLS route-distinguisher 10.255.255.1:1001
set routing-instances VPLS vrf-target target:65100:1001
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface gr-0/0/0.0
set protocols bgp group IBGP type internal
set protocols bgp group IBGP local-address 10.255.255.1
set protocols bgp group IBGP local-as 65100
```

```
set protocols bgp group IBGP neighbor 10.255.255.2 family inet any
set protocols bgp group IBGP neighbor 10.255.255.2 family inet-vpn any
set protocols bgp group IBGP neighbor 10.255.255.2 family l2vpn signaling
set protocols bgp tcp-mss 1200
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols mpls interface gr-0/0/0.0
set routing-options static route 172.16.23.0/30 next-hop 172.16.13.2
set routing-options router-id 10.255.255.1
```

The configuration for the SRX2 (PE2) device:

```
set system host-name SRX2
set security ike policy standard mode main
set security ike policy standard proposal-set standard
set security ike policy standard pre-shared-key ascii-text "$9$Ahg6tORhclvMXREdb2gJZ"
set security ike gateway srx-1 ike-policy standard
set security ike gateway srx-1 address 172.16.13.1
set security ike gateway srx-1 external-interface ge-0/0/1.0
set security ipsec policy standard proposal-set standard
set security ipsec vpn ipsec-vpn-1 bind-interface st0.0
set security ipsec vpn ipsec-vpn-1 df-bit clear
set security ipsec vpn ipsec-vpn-1 ike gateway srx-1
set security ipsec vpn ipsec-vpn-1 ike ipsec-policy standard
set security ipsec vpn ipsec-vpn-1 establish-tunnels immediately
set security policies from-zone Internet to-zone Internet policy Internet match source-address
any
set security policies from-zone Internet to-zone Internet policy Internet match destination-
address any
set security policies from-zone Internet to-zone Internet policy Internet match application any
set security policies from-zone Internet to-zone Internet policy Internet then permit
set security zones security-zone Internet host-inbound-traffic system-services all
set security zones security-zone Internet host-inbound-traffic protocols all
set security zones security-zone Internet interfaces ge-0/0/1.0
set security zones security-zone Internet interfaces gr-0/0/0.0
set security zones security-zone Internet interfaces lo0.0
set security zones security-zone Internet interfaces st0.0
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 mtu 4000
set interfaces ge-0/0/0 description L3VPN
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet filter input packet-mode-inet
```

```
set interfaces ge-0/0/0 unit 10 family inet address 192.168.1.1/24
set interfaces gr-0/0/0 description "MPLS core facing interface"
set interfaces gr-0/0/0 unit 0 tunnel source 172.16.0.2
set interfaces gr-0/0/0 unit 0 tunnel destination 172.16.0.1
set interfaces gr-0/0/0 unit 0 family inet mtu 9000
set interfaces gr-0/0/0 unit 0 family inet address 172.16.255.2/30
set interfaces gr-0/0/0 unit 0 family mpls mtu 9000
set interfaces gr-0/0/0 unit 0 family mpls filter input packet-mode
set interfaces ge-0/0/1 description Internet
set interfaces ge-0/0/1 mtu 1514
set interfaces ge-0/0/1 unit 0 family inet address 172.16.23.1/30
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 mtu 1522
set interfaces ge-0/0/2 encapsulation vlan-vpls
set interfaces ge-0/0/2 unit 11 description VPLS
set interfaces ge-0/0/2 unit 11 encapsulation vlan-vpls
set interfaces ge-0/0/2 unit 11 vlan-id 512
set interfaces lo0 unit 0 family inet address 10.255.255.2/32
set interfaces st0 unit 0 family inet mtu 9178
set interfaces st0 unit 0 family inet address 172.16.0.2/30
set firewall family inet filter packet-mode-inet term all-traffic then packet-mode
set firewall family inet filter packet-mode-inet term all-traffic then accept
set firewall family mpls filter packet-mode term all-traffic then packet-mode
set firewall family mpls filter packet-mode term all-traffic then accept
set routing-instances L3VPN routing-options auto-export
set routing-instances L3VPN interface ge-0/0/0.10
set routing-instances L3VPN instance-type vrf
set routing-instances L3VPN route-distinguisher 10.255.255.2:1000
set routing-instances L3VPN vrf-target target:65100:1000
set routing-instances L3VPN vrf-table-label
set routing-instances VPLS protocols vpls site 2 interface ge-0/0/2.11
set routing-instances VPLS protocols vpls site 2 site-identifier 2
set routing-instances VPLS protocols vpls no-tunnel-services
set routing-instances VPLS protocols vpls mac-tlv-receive
set routing-instances VPLS protocols vpls mac-tlv-send
set routing-instances VPLS interface ge-0/0/2.11
set routing-instances VPLS instance-type vpls
set routing-instances VPLS route-distinguisher 10.255.255.2:1001
set routing-instances VPLS vrf-target target:65100:1001
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface gr-0/0/0.0
set protocols bgp group IBGP type internal
```

```
set protocols bgp group IBGP local-address 10.255.255.2
set protocols bgp group IBGP local-as 65100
set protocols bgp group IBGP neighbor 10.255.255.1 family inet any
set protocols bgp group IBGP neighbor 10.255.255.1 family inet-vpn any
set protocols bgp group IBGP neighbor 10.255.255.1 family l2vpn signaling
set protocols bgp tcp-mss 1200
set protocols ldp interface gr-0/0/0.0
set protocols ldp interface lo0.0
set protocols mpls interface gr-0/0/0.0
set routing-options static route 172.16.13.0/30 next-hop 172.16.23.2
set routing-options router-id 10.255.255.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide for Junos OS.

To fragment the MPLS frame and reassemble the packet:

1. Configure the physical Interfaces.

```
[edit interfaces]
user@SRX1# set ge-0/0/0 description L3VPN
user@SRX1# set ge-0/0/0 mtu 4000
user@SRX1# set ge-0/0/0 unit 10 vlan-id 10
user@SRX1# set ge-0/0/0 unit 10 family inet filter input packet-mode-inet
user@SRX1# set ge-0/0/0 unit 10 family inet address 192.168.0.1/24
user@SRX1# set ge-0/0/1 description Internet
user@SRX1# set ge-0/0/1 mtu 1514
user@SRX1# set ge-0/0/1 unit 0 family inet address 172.16.13.1/30
user@SRX1# set ge-0/0/2 description VPLS
user@SRX1# set ge-0/0/2 flexible-vlan-tagging
user@SRX1# set ge-0/0/2 mtu 1522
user@SRX1# set ge-0/0/2 encapsulation vlan-vpls
user@SRX1# set ge-0/0/2 unit 11 encapsulation vlan-vpls
user@SRX1# set ge-0/0/2 unit 11 vlan-id 512
```

2. Configure the logical Interfaces.

```
[edit interfaces]
user@SRX1# set gr-0/0/0 unit 0 description "MPLS core facing interface"
user@SRX1# set gr-0/0/0 unit 0 tunnel source 172.16.0.1
user@SRX1# set gr-0/0/0 unit 0 tunnel destination 172.16.0.2
user@SRX1# set gr-0/0/0 unit 0 family inet mtu 9000
user@SRX1# set gr-0/0/0 unit 0 family inet address 172.16.255.1/30
user@SRX1# set gr-0/0/0 unit 0 family mpls mtu 9000
user@SRX1# set gr-0/0/0 unit 0 family mpls filter input packet-mode
user@SRX1# set lo0 unit 0 family inet address 10.255.255.1/32
user@SRX1# set st0 unit 0 family inet mtu 9178
user@SRX1# set st0 unit 0 family inet address 172.16.0.1/30
```

3. Configure the firewall filters that are used to configure interfaces to work with packet mode.

```
[edit firewall]
user@SRX1# set family inet filter packet-mode-inet term all-traffic then packet-mode
user@SRX1# set family inet filter packet-mode-inet term all-traffic then accept
user@SRX1# set family mpls filter packet-mode term all-traffic then packet-mode
user@SRX1# set family mpls filter packet-mode term all-traffic then accept
```

NOTE: If you are configuring a Layer 2 Circuit you must also add a filter to evoke packet mode on the CE-facing interface under family CCC:

```
set firewall family ccc filter packet-mode-ccc term all-traffic then packet-mode
set firewall family ccc filter packet-mode-ccc term all-traffic then accept
```

4. Configure the IKE and IPsec policies.

```
[edit security]
user@SRX1# set ike policy standard mode main
user@SRX1# set ike policy standard proposal-set standard
user@SRX1# set ike policy standard pre-shared-key ascii-text "$9$1OsIclKvL7NblegoGUHk"
user@SRX1# set ike gateway srx-2 ike-policy standard
user@SRX1# set ike gateway srx-2 address 172.16.23.1
user@SRX1# set ike gateway srx-2 external-interface ge-0/0/1.0
user@SRX1# set ipsec policy standard proposal-set standard
```

```
user@SRX1# set ipsec vpn ipsec-vpn-1 bind-interface st0.0
user@SRX1# set ipsec vpn ipsec-vpn-1 df-bit clear
user@SRX1# set ipsec vpn ipsec-vpn-1 ike gateway srx-2
user@SRX1# set ipsec vpn ipsec-vpn-1 ike ipsec-policy standard
user@SRX1# set ipsec vpn ipsec-vpn-1 establish-tunnels immediately
```

NOTE: To keep the focus on fragmentation over IPsec we use the default cypher in this example (3DES-CBC). For increased performance and security consider using a newer cypher, such as AES-GCM-256. see encryption-algorithm (Security IKE)

5. Configure all noncustomer-facing interfaces in a single security zone and a policy to permit all (intrazone) traffic.

```
[edit security policies from-zone Internet to-zone Internet]
user@SRX1# set policy Internet match source-address any
user@SRX1# set policy Internet match destination-address any
user@SRX1# set policy Internet match application any
user@SRX1# set policy Internet then permit
[edit security zones security-zone Internet]
user@SRX1# set host-inbound-traffic system-services all
user@SRX1# set host-inbound-traffic protocols all
user@SRX1# set interfaces ge-0/0/1.0
user@SRX1# set interfaces gr-0/0/0.0
user@SRX1# set interfaces lo0.0
user@SRX1# set interfaces st0.0
```

6. Configure the OSPF protocol for lo0.0 address distribution, configure IBGP with the inet-vpn and l2vpn families. Also configure MPLS and LDP signaling.

```
[edit protocols]
user@SRX1# set bgp tcp-mss 1200
user@SRX1# set bgp group IBGP type internal
user@SRX1# set bgp group IBGP local-address 10.255.255.1
user@SRX1# set bgp group IBGP local-as 65100
user@SRX1# set bgp group IBGP neighbor 10.255.255.2
user@SRX1# set bgp group IBGP neighbor 10.255.255.2 family inet any
user@SRX1# set bgp group IBGP neighbor 10.255.255.2 family inet-vpn any
user@SRX1# set bgp group IBGP neighbor 10.255.255.2 family l2vpn signaling
user@SRX1# set ospf traffic-engineering
```

```
user@SRX1# set ospf area 0.0.0.0 interface lo0.0
user@SRX1# set ospf area 0.0.0.0 interface lo0.0 passive
user@SRX1# set ospf area 0.0.0.0 interface gr-0/0/0.0
user@SRX1# set mpls interface gr-0/0/0.0
user@SRX1# set ldp interface gr-0/0/0.0
user@SRX1# set ldp interface lo0.0
```

7. Configure the router ID and a static route to the remote end of the WAN link.

```
[edit routing-option]
user@SRX1# set static route 172.16.23.0/30 next-hop 172.16.13.2
user@SRX1# set router-id 10.255.255.1
```

8. Configure two routing instances, one for Layer 3 VPN and the other for the VPLS application.

```
[edit routing-instances]
user@SRX1# set L3VPN instance-type vrf
user@SRX1# set L3VPN route-distinguisher 10.255.255.1:1000
user@SRX1# set L3VPN interface ge-0/0/0.10
user@SRX1# set L3VPN vrf-target target:65100:1000
user@SRX1# set L3VPN vrf-target import target:65100:1000
user@SRX1# set L3VPN vrf-target export target:65100:1000
user@SRX1# set L3VPN vrf-table-label
user@SRX1# set L3VPN routing-options auto-export
user@SRX1# set VPLS instance-type vpls
user@SRX1# set VPLS interface ge-0/0/2.11
user@SRX1# set VPLS route-distinguisher 10.255.255.1:1001
user@SRX1# set VPLS vrf-target target:65100:1001
user@SRX1# set VPLS protocols vpls no-tunnel-services
user@SRX1# set VPLS protocols vpls site 1 site-identifier 1
user@SRX1# set VPLS protocols vpls site 1 interface ge-0/0/2.11
user@SRX1# set VPLS protocols vpls mac-tlv-receive
user@SRX1# set VPLS protocols vpls mac-tlv-send
```

Results

Display the results of the configuration:

```
user@SRX1> show configuration
security {
    ike {
        policy standard {
            mode main;
            proposal-set standard;
            pre-shared-key ascii-text "$9$1OsIclKvL7NblegoGUHk"; ## SECRET-DATA
        }
        gateway srx-2 {
            ike-policy standard;
            address 172.16.23.1;
            external-interface ge-0/0/1.0;
        }
    }
    ipsec {
        policy standard {
            proposal-set standard;
        }
        vpn ipsec-vpn-1 {
            bind-interface st0.0;
            df-bit clear;
            ike {
                gateway srx-2;
                ipsec-policy standard;
            }
            establish-tunnels immediately;
        }
    }
    policies {
        from-zone Internet to-zone Internet {
            policy Internet {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
```

```
                }
            }
        }
    }
    zones {
        security-zone Internet {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/1.0;
                gr-0/0/0.0;
                lo0.0;
                st0.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        vlan-tagging;
        mtu 4000;
        unit 10 {
            description L3VPN;
            vlan-id 10;
            family inet {
                filter {
                    input packet-mode-inet;
                }
                address 192.168.0.1/24;
            }
        }
    }
    gr-0/0/0 {
        unit 0 {
            description "MPLS core facing interface";
            tunnel {
                source 172.16.0.1;
```

```
            destination 172.16.0.2;
        }
        family inet {
            mtu 9000;
            address 172.16.255.1/30;
        }
        family mpls {
            mtu 9000;
            filter {
                input packet-mode;
            }
        }
    }
}
ge-0/0/1 {
    description Internet;
    mtu 1514;
    unit 0 {
        family inet {
            address 172.16.13.1/30;
        }
    }
}
ge-0/0/2 {
    flexible-vlan-tagging;
    mtu 1522;
    encapsulation vlan-vpls;
    unit 11 {
        description VPLS;
        encapsulation vlan-vpls;
        vlan-id 512;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.255.1/32;
        }
    }
}
st0 {
    unit 0 {
        family inet {
```

```
                mtu 9178;
                address 172.16.0.1/30;
            }
        }
    }
}
firewall {
    family inet {
        filter packet-mode-inet {
            term all-traffic {
                then {
                    packet-mode;
                    accept;
                }
            }
        }
    }
    family mpls {
        filter packet-mode {
            term all-traffic {
                then {
                    packet-mode;
                    accept;
                }
            }
        }
    }
}
routing-instances {
    L3VPN {
        routing-options {
            auto-export;
        }
        interface ge-0/0/0.10;
        instance-type vrf;
        route-distinguisher 10.255.255.1:1000;
        vrf-target {
            target:65100:1000;
            import target:65100:1000;
            export target:65100:1000;
        }
        vrf-table-label;
    }
```

```
    VPLS {
        protocols {
            vpls {
                site 1 {
                    interface ge-0/0/2.11;
                    site-identifier 1;
                }
                no-tunnel-services;
                mac-tlv-receive;
                mac-tlv-send;
            }
        }
        interface ge-0/0/2.11;
        instance-type vpls;
        route-distinguisher 10.255.255.1:1001;
        vrf-target target:65100:1001;
    }
}
protocols {
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0 {
                passive;
            }
            interface gr-0/0/0.0;
        }
    }
    bgp {
        group IBGP {
            type internal;
            local-address 10.255.255.1;
            local-as 65100;
            neighbor 10.255.255.2 {
                family inet {
                    any;
                }
                family inet-vpn {
                    any;
                }
                family l2vpn {
                    signaling;
                }
```

```
            }
        }
        tcp-mss 1200;
    }
    ldp {
        interface gr-0/0/0.0;
        interface lo0.0;
    }
    mpls {
        interface gr-0/0/0.0;
    }
}
routing-options {
    static {
        route 172.16.23.0/30 next-hop 172.16.13.2;
    }
    router-id 10.255.255.1;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That the Physical and Logical Interfaces Are Up

Purpose

Verify that the physical and logical interfaces are up on the device.

Action

From operational mode on the SRX Series Services Gateway, enter the `show interfaces terse` command.

```
user@SRX1> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.10             up    up   inet     192.168.0.1/24
ge-0/0/0.32767          up    up
gr-0/0/0                up    up
gr-0/0/0.0              up    up   inet     172.16.255.1/30
                                   mpls
ip-0/0/0                up    up
lsq-0/0/0               up    up
lt-0/0/0                up    up
mt-0/0/0                up    up
sp-0/0/0                up    up
sp-0/0/0.0              up    up   inet
                                   inet6
sp-0/0/0.16383          up    up   inet
ge-0/0/1                up    up
ge-0/0/1.0              up    up   inet     172.16.13.1/30
ge-0/0/2                up    up
ge-0/0/2.11             up    up   vpls
ge-0/0/2.32767          up    up
dsc                     up    up
fti0                    up    up
fxp0                    up    up
fxp0.0                  up    up   inet     10.54.5.56/19
gre                     up    up
ipip                    up    up
irb                     up    up
lo0                     up    up
lo0.0                   up    up   inet     10.255.255.1        --> 0/0
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
```

```
lo0.16385              up    up   inet     10.0.0.1            --> 0/0
                                           10.0.0.16           --> 0/0
                                           128.0.0.1           --> 0/0
                                           128.0.0.4           --> 0/0
                                           128.0.1.16          --> 0/0
lo0.32768              up    up
lsi                    up    up
lsi.0                  up    up   inet
                                  iso
                                  inet6
lsi.1048576            up    up   vpls
. . .
<some output removed for brevity>
```

Meaning

The output of the `show interfaces terse` command shows that all physical and logical interfaces used in this configuration are operational.

Verifying IPsec Security Associations

Purpose

Verify that the IKE and IPsec security associations are up on the device.

Action

From operational mode on the SRX Series Services Gateway, enter the `show security ike security-association` and `show security ipsec security-association` commands.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
6699112 UP     2a5d1a37e5bd0cd1  09880f53cdbb35bb  Main           172.16.23.1


user@SRX1>  show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1   f1396d7e 1868/ unlim  -   root 500   172.16.23.1
  >131073 ESP:3des/sha1   ff799c04 1868/ unlim  -   root 500   172.16.23.1
```

Meaning

The output shows the expected Up state for the IKE session and that an IPsec tunnel is successfully established.

Verifying OSPF and BGP

Purpose

Verify that OSPF and BGP are operating correctly over the GRE tunnel. Recall that the GRE tunnel is in turn routed over the IPsec tunnel verified in the previous step. Proper OSPF/BGP operation in this example indirectly verifies that traffic is able to pass over the GRE (and then the IPsec) tunnel. If desired, you can ping the GRE endpoint for added verification.

Action

From operational mode on the SRX Series Services Gateway, enter the `show ospf neighbor` and `show bgp summary` commands.

```
user@SRX1> show ospf neighbor
Address          Interface              State           ID               Pri  Dead
172.16.255.2     gr-0/0/0.0             Full            10.255.255.2     128    33


user@SRX1> show bgp summary
  Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       0          0          0          0          0          0
inet.2
                       0          0          0          0          0          0
bgp.l3vpn.0
                       1          1          0          0          0          0
bgp.l3vpn.2
                       0          0          0          0          0          0
bgp.l2vpn.0
                       1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
```

```
10.255.255.2          65100        988        988       0       1      7:21:03 Establ
  inet.0: 0/0/0/0
  inet.2: 0/0/0/0
  bgp.l3vpn.0: 1/1/1/0
  bgp.l3vpn.2: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  L3VPN.inet.0: 1/1/1/0
  VPLS.l2vpn.0: 1/1/1/0
```

Meaning

The output confirms the expected OSPF neighbor state of `full`. This OSPF neighbor is estanlished over the GRE interface. Given OSPF is operational, you expect that the local SRX has learned the route to the remote SRX's loopback address. This route allows the loopback based IBGP peering session to establish (over the GRE tunnel). The output of the `show bgp summary` command confirms the BGP session is in the established state, and that it is exchanging both L3VPN and L2VPN routes.

Verifying LDP Operation

Purpose

Verify that LDP is operating correctly over the GRE tunnel. LDP functions as the MPLS signaling protocol in this example.

Action

From operational mode on the SRX Series Services Gateway, enter the `show ldp neighbor` and `show ldp session` commands.

```
user@SRX1> show ldp neighbor
Address                            Interface       Label space ID     Hold time
172.16.255.2                       gr-0/0/0.0      10.255.255.2:0       12



user@SRX1>  show ldp session
  Address                          State       Connection  Hold time  Adv. Mode
10.255.255.2                       Operational Open          28         DU
```

Meaning

The output confirms the expected LDP neighbor relationship over the GRE interface. The output of the `show ldp session` command confirms successful session establishment to the remote SRX device's loopback address. This allows LDP to exchange transport labels that in turn support MPLS forwarding for the VPN clients.

Verifying The VPLS Connection

Purpose

Verify that the VPLS connection is in an up state.

Action

From operational mode on the SRX Series Services Gateway, enter the `show vpls connections` command.

```
user@SRX1> show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch           HS -- Hot-standby Connection

Legend for interface status
```

```
Up -- operational
Dn -- down

Instance: VPLS
Edge protection: Not-Primary
  Local site: 1 (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Aug 25 07:52:38 2021           1
      Remote PE: 10.255.255.2, Negotiated control-word: No
      Incoming label: 262146, Outgoing label: 262145
      Local interface: lsi.1048578, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls VPLS local site 1 remote site 2
      Flow Label Transmit: No, Flow Label Receive: No
```

Meaning

The output shows the expected `Up` state for the VPLS connection. With the connection operational, the VPN client devices should be able to pass traffic.

Verifying End-to-End VPLS Connectivity for Large Packets with DNF Set

Purpose

Verify that the Layer 2 VPLS client devices are able to send 1500 byte frames with the DNF bit set. Because this is a Layer 2 service, fragmentation is not possible. As a result the DNF bit operates end-to-end. Recall that with the configuration in this example, such a setting results in the ingress SRX device fragmenting the IPsec packet *after* the traffic has been encrypted (post-fragmentation). The post-fragmentation occurs as the traffic egresses the WAN facing ge-0/0/1 interface.

Post-fragmentation forces the remote SRX device to reassemble the packet before it can perform decryption, which can impact forwarding performance for encrypted traffic. This is the expected behavior when the `df-bit clear` option is used. Demonstration this behavior is the reason for this NCE. The other `df-bit` options, namely `df-bit copy` and `df-bit set`, result in packet discard and generation of an ICMP error message for VPN packets that exceed the WAN MTU when the DNF bit is set by the VPN client.

Action

From operational mode on VPLS Host1, ping VPLS Host2 in a manner that generates a 1500 byte IP packet with the DNF bit set. When this traffic has the MPLS, GRE, and IPsec overhead added it exceeds the outgoing WAN interface's MTU. Given that pre-fragmentation is blocked by virtue of this being a

Layer 2 service (or in the case of the L3VPN client, by setting the DNF bit), such a packet forces post-fragmentation based on the setting of the `df-bit clear` option

The configuration and operation of the VPN client devices are outside the scope of this example. For testing, a MX router is used to act as the VPN clients. As a result the ping command demonstrated is based on the Junos CLI.

```
user@vpls-host1> ping 192.168.2.102 size 1472 do-not-fragment count 2

PING 192.168.2.102 (192.168.2.102): 1472 data bytes
1480 bytes from 192.168.2.102: icmp_seq=0 ttl=64 time=23.045 ms
1480 bytes from 192.168.2.102: icmp_seq=1 ttl=64 time=5.342 ms

--- 192.168.2.102 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.342/14.194/23.045/8.852 ms
```

Meaning

The output shows the pings succeed. The 1480 bytes of echo traffic results in a 1500 byte IP packet when the 20 byte IP header is added. Thus, the results confirm the VPLS client device can exchange 1,500 byte packets over a WAN link with a 1,500 byte MTU, despite the encapsulation overhead. Recall that because this is a Layer 2 service, fragmentation is not possible and the DNF bit operates end-to-end. Using the DNF bit is significant when testing the L3VPN client, however, because the PE device is able to fragment IP traffic.

Verifying IP Fragmentation on the Outgoing Interface

Purpose

Verify that VPLS client traffic that exceeds the WAN MTU is fragmented on the outgoing ge-0/0/1.0 interface. Timing is important in this step because background OSPF, LDP, and BGP traffic causes the ge-0/0/0.0 interface counters to increment. The goal is to generate 100 1,500 byte packets from the VPLS host and then quickly compare the IPsec and interface statistics to confirm that approximately twice as many packets are seen on the outgoing WAN interface when compared to the counts on the IPsec tunnel.

Action

From operational mode on the SRX Series Services Gateway, clear both the IPsec and interface statistics with the `clear interfaces statistics all` and `clear security ipsec statistics` commands. Then generate 100 rapid pings with a 1,500 byte packet size between the VPLS endpoints. When the pings complete, display packet counts for the IPsec tunnel and the ge-0/0/1 interface with the `show interfaces ge-0/0/1 detail` and `show security ipsec statistics` commands.

```
user@SRX1> clear interfaces statistics all
user@SRX1> clear interfaces statistics all
```

Generate 100 rapid pings with a packet size of 1,500 bytes between the VPLS endpoints. This is not shown for brevity. Refer to the command in the previous step. Not shown here for brevity.

```
user@SRX1> show interfaces ge-0/0/1 detail
Physical interface: ge-0/0/1, Enabled, Physical link is Up
  Interface index: 136, SNMP ifIndex: 509, Generation: 139
  Description: Internet
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex, Speed: 10Gbps,
BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 56:04:19:00:3a:7b, Hardware address: 56:04:19:00:3a:7b
  Last flapped   : 2021-08-27 12:17:01 PDT (01:27:43 ago)
  Statistics last cleared: 2021-08-27 13:44:28 PDT (00:00:16 ago)
  Traffic statistics:
   Input  bytes  :              163440                    0 bps
   Output bytes  :              162000                    0 bps
   Input  packets:                 210                    0 pps
   Output packets:                 200                    0 pps
  Egress queues: 8 supported, 4 in use
. . .

user@SRX1> show security ipsec statistics
ESP Statistics:
  Encrypted bytes:          161896
  Decrypted bytes:          155722
  Encrypted packets:           113
  Decrypted packets:           112
. . .
```

Meaning

The output of the `show interfaces ge-0/0/1.0 detail` command shows that over200 packets have been sent and received. In contrast, the IPsec statistics confirm a count of around 100 packets. This confirms that each packet sent by the VPLS client was fragmented on the WAN-facing ge-0/0/1.0 interface.

Verifying The L3VPN

Purpose

Verify L3VPN Operation.

Action

From operational mode on the SRX Series Services Gateway, display the route to the remote L3VPN subnet with the `show route` command. Then generate pings to the remote L3VPN endpoint to verify connectivity.

```
user@SRX1> show route 192.168.1.0/24
L3VPN.inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.0/24     *[BGP/170] 01:05:44, localpref 100, from 10.255.255.2
                      AS path: I, validation-state: unverified
                    >  via gr-0/0/0.0, Push 16

bgp.l3vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.255.2:1000:192.168.1.0/24
                   *[BGP/170] 01:05:44, localpref 100, from 10.255.255.2
                      AS path: I, validation-state: unverified
                    >  via gr-0/0/0.0, Push 16
```

Test connectivity from the local SRX to the remote VPN endpoint:

```
user@SRX1> ping 192.168.1.101 routing-instance L3VPN count 2
PING 192.168.1.101 (192.168.1.101): 56 data bytes
64 bytes from 192.168.1.101: icmp_seq=0 ttl=63 time=3.485 ms
64 bytes from 192.168.1.101: icmp_seq=1 ttl=63 time=3.412 ms
```

```
--- 192.168.1.101 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.412/3.449/3.485/0.036 ms
```

NOTE: In this configuration a ping from the local SRX to the local L3VPN client does not succeed. This relates to the use of packet mode and the lack of security zones for the VPN interfaces. As shown above, you are able to ping from the local SRX to the remote L3VPN destinations. Though not shown, a ping generated from the local L3VPN client to the local PE VRF interface is expected to succeed.

Test end-to-end connectivity for the L3VPN. Generate jumbo pings between L3VPN client endpoints. Recall that the L3VPN client is configured with a 4k MTU in this example. Once again we use a MX router to fill in for the L3VPN client, so Junos ping syntax is used:

```
user@l3vpn1> ping 192.168.1.101 size 3000 do-not-fragment count 2
PING 192.168.1.101 (192.168.1.101): 3000 data bytes
3008 bytes from 192.168.1.101: icmp_seq=0 ttl=62 time=5.354 ms
3008 bytes from 192.168.1.101: icmp_seq=1 ttl=62 time=5.607 ms

--- 192.168.1.101 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.354/5.481/5.607/0.126 ms
```

Meaning

The output shows the route to the remote L3VPN client is correctly learned via BGP, and that it points to the GRE interface with an MPLS label operation. The results from ping testing confirm expected connectivity for the L3VPN even when sending 3,000 + byte pings with the DNF bit set.

RELATED DOCUMENTATION

Simplified MPLS Through IPsec over 1500-byte Media Overview | 2

Example: Configuring MPLS over GRE with IPsec Fragmentation and Reassembly | 3