INTRODUCTION TO GROUP VPN

Configuring Group VPN on Branch SRX Series Services Gateways

Copyright © 2011, Juniper Networks, Inc.

Table of Contents

Table of Figures

Copyright © 2011, Juniper Networks, Inc.

Introduction

Group VPN is a new category of VPN that eliminates the need for point-to-point VPN tunnels in a mesh architecture. Group Encrypted Transport is built on standards-based technologies that integrate routing and encryption together in the network. Secure group members are managed through the Group Domain of Interpretation standard (GDOI).

Traditional IPsec VPN deployments tackle the problem of securing traffic between gateways in the network by creating an overlay network based on the use of point-to-point tunnels. Traffic carried over these tunnels is normally encrypted and authenticated in order to provide data integrity and confidentiality. The GDOI solution takes a different approach by disassociating the encryption and authentication problem from the transport. By doing this, GDOI-based solutions can provide a way to encrypt branch-to-branch communications without the need to configure branch-to-branch tunnels. The obvious downside to this approach is that without branch-to-branch tunnels, the transport network must be able to natively route the branch-to-branch traffic without any tunneling. At the time of this writing, several solutions to this transport problem are being considered, such as the use of dynamic GRE tunnels or BGP extensions.

GDOI alleviates the need to configure branch-to-branch tunnel endpoints. A key server distributes keys and policies to all registered and authenticated member routers. By distributing policies from a centralized point and by sharing the same group security association (the entire group has a single Phase 2 IPsec SA (Security Association)) with authenticated group members, key distribution and management are greatly simplified.

Group VPN is client/server architecture. All members have a unique Phase 1 IKE SA with the key server. Hence, if there are n members, there is a total of n Phase 1 IKE SAs. However, the entire group shares a single Phase 2 SA. One important difference between Group VPN and traditional VPNs is that the external IP header in Group VPN is an exact copy of the IP header of the original packet within the ESP header versus the external IP header in a traditional VPN contains the IP addresses of the VPN gateways.

Platform Support

The Group VPN feature set is supported on Juniper Networks® SRX Series Services Gateways for the branch, working in standalone mode. The following are the lists of supported devices and minimum software versions required.

Hardware Requirements

- Branch SRX Series Services Gateways (SRX100, SRX200 line, and SRX650)

Software Requirements

- Juniper Networks Junos® operating system Release 10.2r2 or later

Group VPN Advantages

- It is standards based.
- It provides instantaneous, large-scale, any-to-any IP connectivity using a group IPsec security paradigm.
- It takes advantage of underlying IP VPN routing infrastructure and does not require an overlay routing control plane.
- It seamlessly integrates with multicast infrastructures without the multicast replication issues typically seen in traditional tunnel-based IPsec solutions.
- It preserves the IP source and destination addresses during the IPsec encryption and encapsulation process. Therefore, Group VPN integrates very well with features such as QoS and traffic engineering.
- It eases scaling issues as there is only one Phase 2 SA.

To provide a true full mesh or even dense partial mesh of connectivity, traditional tunnel-based solutions require the provisioning of a complex connectivity mesh. Such a complex mesh not only has higher processor and memory requirements, but it is difficult to provision, troubleshoot, and manage.

Traditional point-to-point IPsec tunneling solutions suffer from multicast replication issues because multicast replication must be performed before tunnel encapsulation and encryption at the IPsec CE (customer edge) router closest to the multicast source. Multicast replication cannot be performed in the provider network because encapsulated multicasts appear to the core network as unicast data.

Copyright © 2011, Juniper Networks, Inc.

Group Encrypted Transport VPN (Group VPN) introduces the concept of a trusted group to eliminate point-to-point tunnels and their associated overlay routing. All group members (GMs) share a common security association (SA), also known as a group SA. This enables GMs to decrypt traffic that was encrypted by any other GM. (Note that the IPsec CE router acts as a GM.) In Group VPN networks, there is no need to negotiate point-to- point IPsec tunnels between the members of a group, because Group VPN is tunnel-less.

Technology Overview

Group encryption introduces some new concepts around IPsec, expanding on the basic support in order to provide client-to-client encryption without the need for spoke-to-spoke negotiation. This section details some of these new concepts.

GDOI

The GDOI protocol described in RFC 3547 is used to distribute a set of cryptographic keys and policies to a group of devices. The communication in the GDOI protocol is protected by a Phase 1 key between each member and the server. GDOI introduces two different encryption keys.

- Key encryption key (KEK)—used to secure the control plane
- Traffic encryption key (TEK)—used to secure the data plane

As with standard IPsec, all keys have a lifetime and have to be rekeyed. The keys distributed via GDOI are group keys and are used by the entire group.

IP Header Preservation

In traditional IPsec, tunnel endpoint addresses are used as new packet source and destination. The packet is then routed over the IP infrastructure, using the encrypting gateway source IP address and the decrypting gateway destination IP address. In the case of Group VPN, IPsec protected data packets encapsulate the original source and destination packet addresses of the host in the outer IP header to preserve the IP address. The biggest advantage of tunnel header preservation is the ability to route encrypted packets using the underlying network routing infrastructure. Because tunnel header preservation is combined with group SAs, multicast replication can be offloaded to the provider network. Because every GM shares the same SA, the IPsec router closest to the multicast source does not need to replicate packets to all its peers—and it is no longer subject to multicast replication issues seen in traditional IPsec solutions.

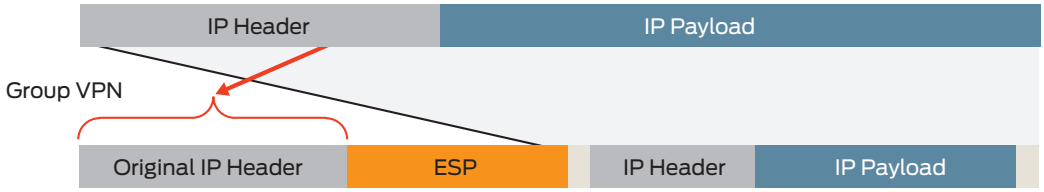

Figure 1. Group VPN IP header

Key Server

A key server is a device used for creating and maintaining the Group VPN control plane. All encryption policies—such as interesting traffic, encryption protocols, security association, rekey timers, and so on—are centrally defined on the key server and are pushed down to all group members at registration time. GMs authenticate with the KS using IKE Phase 1 and then download the encryption policies and keys required for Group VPN operation. The KS is also responsible for refreshing and distributing the keys. Unlike traditional IPsec, interesting traffic defined on the KS is downloaded to every GM, whether or not the GM owns that network.

Copyright © 2011, Juniper Networks, Inc.

Group Member

A group member is a device that is responsible for the actual encryption and decryption of data traffic. A GM is only configured with IKE Phase 1 parameters and KS/group information. As mentioned before, encryption policies are defined centrally on the KS and downloaded to the GM at the time of registration. Based on these downloaded policies, the GM determines whether traffic needs to be encrypted or decrypted and what keys to use.

Group SA

Unlike traditional IPsec encryption solutions, Group VPN uses the concept of group SA. All members in the Group VPN group can communicate with each other using a common encryption policy and a shared SA. With a common encryption policy and a shared SA, there is no need to negotiate IPsec between GMs—this reduces the resource load on the IPsec routers. Traditional GM scalability (number of tunnels and associated SA) does not apply to Group VPN GMs.

Rekey

There are three rekey methods associated with group VPN.

Pull

In the PULL method, the GM requests the group SA and policy from the key server. This request is protected over the IKE SA. No KEK is required for the PULL method.

Unicast Push

In the unicast rekey process, a KS generates a rekey message and sends multiple copies of the message, one copy to each GM. The message that the KS sends is protected over KEK. Upon receiving the rekey message, a GM sends an ACK message to the KS. This ACK mechanism not only ensures that the list of active GMs on the KS is current, but also ensures that the rekey message is sent only to active GMs. A KS can be configured to retransmit a rekey packet to overcome transient defects in the network. If a GM does not acknowledge three consecutive rekeys (retransmissions are considered part of the rekey), the KS removes the GM from its active GM database and stops sending rekey messages to that GM.

Multicast Push

In the multicast rekey process, a KS generates a rekey message and sends one copy of the message to a multicast group address that is predefined in the configuration. The message that the KS sends is protected over KEK. Each GM joins the multicast group at registration time, so each GM receives a copy of the rekey message. Unlike unicast rekey, multicast rekey does not have an ACK mechanism. The KS does not maintain a list of active GMs.

Group Encryption on SRX Series Services Gateways

The following explains the operation of Group VPN on the SRX Series Services Gateways using the aforementioned topology. “Data Center” is the key server while “Black,” ”White,” and “Rainbow” are group members. Each member has a protected/private network behind it. The loopback addresses on each router for server-member peering are used. Routes are propagated using OSPF between all four routers.

Copyright © 2011, Juniper Networks, Inc.

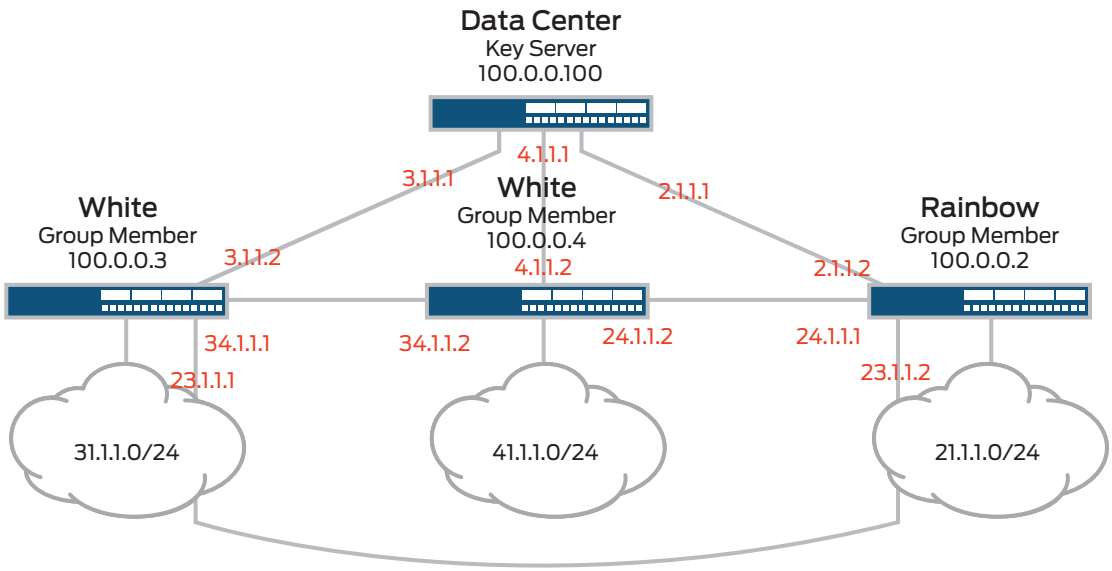

Figure 2. Group VPN topology

Key Server Configuration

The following elements need to be configured on the key server:

1. IKE SA parameters to each of the members
2. IPsec proposal for the group
3. Group Configuration
   a. Group ID
   b. IKE gateways that are part of the group
   c. Policy for interesting traffic
   d. Server-Member Communication (optional, but recommended)

IKE SA Configuration

This section defines the IKE proposal, IKE policy, and gateway definitions for each of the members. This is very similar to standard IKE defentions—the only difference is that it is configured under the group-vpn hierarchy.

```
root@data-center# show security group-vpn server ike
proposal ike_proposal {
    authentication-method pre-shared-keys;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_policy {
    mode main;
    proposals ike_proposal;
    pre-shared-key ascii-text “$9$F6aR6CuRhr8X-O1X-VwaJ369”; ## SECRET-DATA
}
gateway black {
    ike-policy ike_policy;
    address 100.0.0.3;
}
gateway white {
```


Copyright © 2011, Juniper Networks, Inc.

```
    ike-policy ike_policy;
    address 100.0.0.4;
}
gateway rainbow {
    ike-policy ike_policy;
    address 100.0.0.2;
}
root@data-center#
IPsec Proposal
This section defines the IPsec proposal that is used by a group. A group can have
one or more SAs/TEK, and each SA has only one IPsec proposal.

root@data-center# show security group-vpn server ipsec
proposal sa-prop {
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 240;
}
root@data-center#
```

Group Configuration

This section defines the group ID, the members of the group, the IP address of the key server, and all SA (Security Association) parameters. Each SA contains an IPsec proposal as well as a policy for encrypting/decrypting interesting traffic. A group needs a minimum of one SA definition. One SA is the most common deployment. Within each SA stanza you can define multiple policies for interesting traffic. Each SA corresponds to a single TEK for the group. If you define n SAs. then the group has n TEKs .

```
root@data-center# show security group-vpn server group g1
group-id 1;
ike-gateway black;
ike-gateway white;
ike-gateway rainbow;
anti-replay-time-window 100;
server-address 100.0.0.100;
ipsec-sa sa {
    proposal sa-prop;
    match-policy dynamic1 {
        source 31.1.1.0/24;
        destination 41.1.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
    }
    match-policy dynamic2 {
        source 41.1.1.0/24;
        destination 31.1.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
    }

    match-policy dynamic3 {
        source 21.1.1.0/24;
        destination 31.1.1.0/24;
```

Copyright © 2011, Juniper Networks, Inc.

```
        source-port 0;
        destination-port 0;
        protocol 0;
    }
}
[edit]
root@data-center#
```

The previous configuration defines a group with group-id "1" whose members are "black," "white," and "rainbow." An anti-replay-time-window of 100 seconds is also defined. The key server IP address is defined as 100.0.0.100, which is the loopback IP address. This value can be between 60 and 360 seconds. A single SA for this group is then defined. This means a single key is used for encryption and decryption of traffic between all members of this group. Within the SA definition the IPsec proposal "sa-prop" to be used for the group is provided. The policies that govern the selection of interesting traffic for encryption and decryption are also defined. In the previous example, traffic from 31.1.1.0/24 to 41.0.0.0/24, traffic from 41.1.1.0/24 to 31.1.1.0/24, and traffic from 21.1.1.0/24 to 31.1.1.0/24 are to be encrypted/decrypted. Of course, return traffic that matches a session is encrypted/decrypted. Note that traffic from 21.1.1.0/24 to 41.1.1.0/24 is in clear text as it has not been defined as part of the interesting traffic. All the traffic in the previous example is encrypted/decrypted by the same TEK.

You can view the IKE SAs (one for each peer) and group IPsec SA on the key server by using the following CLI command:

```
root@data-center# run show security group-vpn server ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
6550    100.0.0.5       UP     4de435fce449e945  919ff734abc97ec  Main
6549    100.0.0.4       UP     2ac037f7e6c9e0bc  204ee7d2f0fbff72  Main
  6548    100.0.0.3       UP     3430437e6ce7a424  7b83d5a84304001b  Main
 root@data-center#

 root@data-center# run show security group-vpn server ipsec security-associations
Group: g1, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  sa                ESP:aes-128/sha1 992cb0e          86
root@data-center#
```

In case you want different keys for encrypting traffic between 21.1.1.0/24 to 31.1.1.0/24, from 31.1.1.0/24 to 41.1.1.0/24, and 41.1.1.0/24 to 31.1.10/24, your SA definitions need to be modified like the following example. In this case you have two SAs and two TEKs—one for each SA.

```
        ipsec-sa sa {
            proposal sa-prop;
            match-policy dynamic1 {
                source 31.1.1.0/24;
                destination 41.1.0.0/16;
                source-port 0;
                destination-port 0;
                protocol 0;
            }

            match-policy dynamic2 {
                source 41.1.1.0/24;
                destination 31.1.0.0/16;
```


Copyright © 2011, Juniper Networks, Inc.

```
                source-port 0;
                destination-port 0;
                protocol 0;
            }
        }

        ipsec-sa sa2 {
            proposal sa-prop;
            match-policy dynamic3 {
                source 21.1.1.0/24;
                destination 31.1.1.0/24;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
        }
```

In the previous case the group has two IPsec SAs that can be viewed using the following CLI command:

```
root@data-center# run show security group-vpn server ipsec security-associations
Group: g1, Group Id: 1
  Total IPsec SAs: 2
  IPsec SA          Algorithm        SPI              Lifetime
  sa                ESP:aes-128/sha1 992cb0e          86
  sa2               ESP:aes-128/sha1 96d0d680         127
root@data-center#
```

In case you want all traffic between the subnets 21.1.1.0/24, 31.1.1.0/24, and 41.1.1.0/24 to be encrypted/decrypted, you can modify your SA and policy as shown in the following example. For the group-vpn policy it is OK for the source and destination to be the same, unlike the policies used for traditional VPNs.

```
ipsec-sa sa {
            proposal sa-prop;
            match-policy dynamic1 {
                source 0.0.0.0/2;          ##0.0.0.0 to  63.255.255.255
                destination 0.0.0.0/2;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
        }
```

Server-Member Communication

The Server-Member Communication configuration is not mandatory. If not defined it is implied that for rekey, the members pull the SA information from the key server. This message is protected over the IEK SA. The key server is not only responsible for generating keys, but also for refreshing and distributing keys to the group members. Group VPN supports two kinds of rekey messages, unicast and multicast. Whenever the Server-Member Communication is defined for a group, a KEK is established for each group. All rekey messages are protected over KEK.

Copyright © 2011, Juniper Networks, Inc.

Unicast

```
root@data-center# show security group-vpn server group g1 server-member-
communication
communication-type unicast;
retransmission-period 30;
number-of-retransmission 3;
encryption-algorithm aes-256-cbc;
sig-hash-algorithm sha1;
root@data-center#
```

In the configuration for Server-Member Communication you need to define the communication type as “unicast” and define parameters for encryption algorithm, hash algorithm, retransmission period, and number of retransmissions. The key server re-transmits SA/TEK information if it does not receive an acknowledgement from the member. The retransmission period defines the time between each retransmission, and the number of retransmission defines the number of retransmissions after which the server marks the member dead, if it did not receive an acknowledgment.

Multicast

In the multicast rekey process, a KS generates a rekey message and sends one copy of the message to a multicast group address that is predefined in the configuration. Each GM joins the multicast group at registration time, so each GM receives a copy of the rekey message. Unlike unicast rekey, multicast rekey does not have an ACK mechanism. The KS does not maintain a list of active GMs. Multicast rekey uses the same low CPU overhead whether there is one GM in the group or a few thousand. Just like unicast rekey, the KS can be configured to retransmit a multicast rekey packet to overcome transient network defects.

```
root@data-center# ... server group g1 server-member-communication
communication-type multicast;
multicast-group 226.1.1.1;
multicast-outgoing-interface lo0.0;
encryption-algorithm aes-256-cbc;
sig-hash-algorithm sha1;
root@data-center#
```

In the configuration for Server-Member Communication you need to define the communication type as “multicast” and define parameters for encryption algorithm, hash algorithm, multicast group, and multicast outgoing interface. As there is no acknowledgments for multicast rekey, there is no point in configuring retransmission interval and threshold. It is a good idea to configure the key server as RP for the multicast group. It is important to note that the configuration under VPN takes care of all the IGMP joins and leaves, and IGMP does not have to be explicitly configured.

```
root@data-center# show protocols pim
rp {
    local {
        address 100.0.0.100;
    }
}
interface all {
    mode sparse;
}
root@data-center#
```

You can view the KEK security associations using the following CLI command:

```
root@data-center# run show security group-vpn server kek security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  GroupId
6541    0.0.0.0         UP     2540c18fd6c2acba  df668ea522b69d91  1
root@data-center#
```


Copyright © 2011, Juniper Networks, Inc.

Group Member Configuration

The configuration on each group member is minimal. You need to define IKE parameters for the key server and define which group is the member a part of. This group definition is done under the IPsec stanza of the configuration.

It is important to note that we need to specify the interface used to send secured traffic, referred in the configuration as the group-vpn-external-interface. Note that this is NOT necessarily the interface used to reach the key server. This is the interface on which traffic is encrypted/decrypted. Of course, this highlights a characteristic of the Group VPN solution; only one interface can be used to send or receive secured traffic per group. However, because a member can be part of multiple groups on the same key server, this should not impose any severe limitations.

```
root@black# show security group-vpn
member {
    ike {
        proposal ike_proposal {
            authentication-method pre-shared-keys;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy ike_policy {
            mode main;
            proposals ike_proposal;
            pre-shared-key ascii-text "$9$km5FCtOcyKn/yKM8dVqmf"; ## SECRET-DATA
        }
        gateway gateway {
            ike-policy ike_policy;
            address 100.0.0.100;
            local-address 100.0.0.3;
        }
    }
    ipsec {
        vpn gvpn {
            ike-gateway gateway;
            group-vpn-external-interface lo0.0;
            group 1;
        }
    }
}
root@black#
```

You also need to define scope policies. A scope policy is a policy that is associated with a group VPN. It maps to a dynamic policy that is received from the key server. It is important to note that in group VPN the policies to identify interesting traffic are defined on the key server. The dynamic policy is pushed to the group member only once TEK is established. The scope policy is always a superset of the dynamic policy. For example, if you define the source and destinations on the key server as 1.1.1.0/22 and 2.2.2.0/22, you cannot define 1.1.1.0/24 and 2.2.2.0/24 as source and destination in the scope policy. It is common practice to define 0.0.0.0/0 as source, 0.0.0.0/0 as destination, and any protocol as the parameters of a scope policy. You also need to enable dynamic policies.

```
root@black# show security policies
from-zone trust to-zone untrust {
    policy scope1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
```

Copyright © 2011, Juniper Networks, Inc.

```
            permit {
                tunnel {
                    ipsec-group-vpn gvpn;
                }
            }
        }
    }
}
from-zone untrust to-zone trust {
    policy scope1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn gvpn;
                }
            }
        }
    }
}

default-policy {
    permit-all;
}
dynamic-policy {
    enable;
}
```

You can view the IKE/IPsec/KEK keys on the group members using the following CLI:

```
root@black# run show security group-vpn member ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
3720    100.0.0.100     UP     170ffd25fca8bffe  81b0ddb5779225e4  Main
[edit]
root@black# run show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId vsys
  >133955590 100.0.0.100 848   ESP:aes-128/sha1 217a0d31 91/  unlim   1   root
  <133955590 100.0.0.100 848   ESP:aes-128/sha1 217a0d31 91/  unlim   1   root
root@black# run show security group-vpn member kek security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  GroupId
3707    100.0.0.100     UP     4704a141e58f5cc0  ebd5cae1b2875259  1
root@black#
```

You can also view the dynamic policies you received from the key server using the following CLI. These policies are never configured on the group member.

Copyright © 2011, Juniper Networks, Inc.

```
root@black# run show security dynamic-policies
From zone: trust, To zone: untrust
  Policy: scope1-0001, State: enabled, Index: 1048580, Scope Policy: 4, Sequence
number: 1
  Source addresses:
    N/A: 31.1.1.0/24
  Destination addresses:
    N/A: 41.1.0.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: untrust, To zone: trust
  Policy: scope1-0001, State: enabled, Index: 1048581, Scope Policy: 5, Sequence
number: 1
  Source addresses:
    N/A: 31.1.1.0/24
  Destination addresses:
    N/A: 41.1.0.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: trust, To zone: untrust
  Policy: scope1-0002, State: enabled, Index: 2097156, Scope Policy: 4, Sequence
number: 2
  Source addresses:
    N/A: 41.1.1.0/24
  Destination addresses:
    N/A: 31.1.0.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: untrust, To zone: trust
  Policy: scope1-0002, State: enabled, Index: 2097157, Scope Policy: 5, Sequence
number: 2
  Source addresses:
    N/A: 41.1.1.0/24
  Destination addresses:
    N/A: 31.1.0.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: trust, To zone: untrust
  Policy: scope1-0003, State: enabled, Index: 3145732, Scope Policy: 4, Sequence
number: 3
  Source addresses:
    N/A: 21.1.1.0/24
  Destination addresses:
    N/A: 31.1.1.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: untrust, To zone: trust
  Policy: scope1-0003, State: enabled, Index: 3145733, Scope Policy: 5, Sequence
number: 3
  Source addresses:
    N/A: 21.1.1.0/24
  Destination addresses:
    N/A: 31.1.1.0/24
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
```

Copyright © 2011, Juniper Networks, Inc.

Group VPN Data Path

The entire Group VPN data path is on the group members. Once the IPsec SAs/TEK is up and distributed to all group members, the key server distributes Policy to each member to determine which traffic is interesting traffic. The policies distributed to each member is exactly the same. However, the from-zone to-zone combination to which this policy applies depends on the from-zone to-zone pairing in the scope policy defined on the group member. Once the policies have been distributed, a tunnel session with source and destination as 0.0.0.0/0 and 0.0.0.0/0 and protocol any is installed on the member.

```
root@black# run show security flow session tunnel
Session ID: 34575, Policy name: N/A, Timeout: N/A, Valid
  In: 0.0.0.0/62334 --> 0.0.0.0/53282;esp, If: lo0.0, Pkts: 0, Bytes: 0
Total sessions: 1
  root@black#
```

The following explains the Group VPN data path in two sections—Encrypting Side Data Path and Decrypting Side Data Path. Specifically, details are provided about the data path—with simple unicast traffic like a telnet—and how Group VPN deals with multicast later on. Explanations are also provided about the data path, with an example of a telnet that originated on a host 41.1.1.2 to a host 31.1.1.2. It is important to note that the traffic never traverses the key server in the example. In some cases it might, but it is purely based on routing.

Encrypting Side Data Path

1. A telnet is done from 31.1.1.2 to 41.1.1.2.
2. The initial syn packet reaches “black” on the trust zone.
3. A policy lookup is done from the trust zone to the untrust zone, given that the destination is 41.1.1.2 and the route points to interface ge-0/0/10 (interface in untrust zone).
4. The dynamic policy that is pushed down from the key server is hit/matched.
5. Three sessions are created.

```
Session ID: 25911, Policy name: scope1-0002/2097156, Timeout: 1798, Valid
  In: 31.1.1.2/57887 --> 41.1.1.2/23;tcp, If: ge-0/0/1.0, Pkts: 40, Bytes: 2182
  Out: 41.1.1.2/23 --> 31.1.1.2/57887;tcp, If: lo0.0, Pkts: 31, Bytes: 1874

Session ID: 25912, Policy name: N/A, Timeout: N/A, Valid
  In: 31.1.1.2/18788 --> 41.1.1.2/24687;esp, If: ge-0/0/10.0, Pkts: 0, Bytes: 0

Session ID: 25913, Policy name: N/A, Timeout: N/A, Valid
  In: 31.1.1.2/0 --> 41.1.1.2/0;esp, If: lo0.0, Pkts: 0, Bytes: 0
```

6. Session ID “25911” is the plain-text session created for the telnet traffic.
7. Session ID “25912” is called the tunnel session. This is used for decrypting any traffic that is received from 41.1.1.2 via the tunnel. This session is tied to the IPsec SA /SPI.
8. Session ID “25913” is called a shadow tunnel session. This is used to drop any IPsec traffic that does not belong to the corresponding SPI.
9. The packet is then encrypted, and the inner source and destination IPs are copied to the outer header and then sent to the next-hop router via interface ge-0/0/10.
10. The reverse packet is matched using session “25912.”

Copyright © 2011, Juniper Networks, Inc.

Decrypting Side Data Path

1. For the telnet from 41.1.1.2 to 31.1.1.2, the encrypted packet arrives on the untrust interface (ge-0/0/11.0) on "white."
2. This packet matches the 0.0.0.0/xxxxx --> 0.0.0.0/xxxxx session created for the IPsec SA/SPI after the key server's pushed down policy.
3. Three new sessions similar to encrypt side are created.

```
Session ID: 34951, Policy name: scope1-0002/2097157, Timeout: 1798, Valid
  In: 41.1.1.2/49268 --> 31.1.1.2/23;tcp, If: ge-0/0/10.0, Pkts: 40, Bytes: 2182
  Out: 31.1.1.2/23 --> 41.1.1.2/49268;tcp, If: ge-0/0/15.0, Pkts: 31, Bytes: 1874

Session ID: 34952, Policy name: N/A, Timeout: N/A, Valid
  In: 41.1.1.2/58025 --> 31.1.1.2/54331;esp, If: ge-0/0/10.0, Pkts: 0, Bytes: 0

Session ID: 34953, Policy name: N/A, Timeout: N/A, Valid
  In: 41.1.1.2/0 --> 31.1.1.2/0;esp, If: lo0.0, Pkts: 0, Bytes: 0
```

4. Session "34951" is for the plain-text traffic. Session "34952" is the tunnel session, and session "34953" is the shadow tunnel session.
5. The traffic gets decrypted and reaches the host 41.1.1.2, and the reverse traffic matches session "34952" and gets encrypted.

Multicast Traffic

Multicast traffic is handled very similarly to unicast traffic. There is no necessity to fan out multicast traffic before it gets encrypted. The multicast data path can be explained with an example. The same topology is used as the one to explain the unicast example. A host in each of the three networks—21.1.1.0/24, 31.1.10/24, and 41.1.1.0/24—subscribes/joins the multicast group 226.1.1.00. The multicast configuration on "white," "black," and "rainbow" is pretty straightforward as illustrated in the following example.

```
root@rainbow# show protocols igmp

interface all;

root@rainbow# show protocols pim
rp {
    static {
        address 100.0.0.100;
    }
}
interface all;
```

The multicast data path can be explained with an example where the host 31.1.1.2 sends UDP traffic to 226.1.1.100. For this example, all the group members need to identify multicast traffic to 226.1.1.100 as interesting traffic. Hence, the policy on the key server looks like the following example.

```
root@data-center# show security group-vpn server group g1
group-id 1;
ike-gateway black;
ike-gateway white;
ike-gateway rainbow;
anti-replay-time-window 100;
server-address 100.0.0.100;
```

Copyright © 2011, Juniper Networks, Inc.

```
server-member-communication {
    communication-type multicast;
    multicast-group 226.1.1.1;
    multicast-outgoing-interface lo0.0;
    retransmission-period 30;
    number-of-retransmission 3;
    encryption-algorithm aes-256-cbc;
    sig-hash-algorithm sha1;

}

ipsec-sa sa {
    proposal sa-prop;
    match-policy dynamic3 {
        source 0.0.0.0/2;
        destination 226.1.1.100/32;
        source-port 0;
        destination-port 0;
        protocol 0;
    }

    match-policy dynamic4 {
        source 0.0.0.0/2;
        destination 0.0.0.0/2;
        source-port 0;
        destination-port 0;
        protocol 0;
    }
}
```

The first policy, dynamic3, is used to allow any traffic from any of the three subnets to 226.1.1.100, while the second policy, dynamic4, is used to allow unicast traffic between all three subnets. The policy that is pushed from the key server can be viewed as a dynamic policy on each of the group members.

```
root@black# run show security dynamic-policies
From zone: trust, To zone: untrust
  Policy: scope1-0001, State: enabled, Index: 1048580, Scope Policy: 4, Sequence
number: 1
  Source addresses:
    N/A: 0.0.0.0/2
  Destination addresses:
    N/A: 226.1.1.100/32
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: untrust, To zone: trust
  Policy: scope1-0001, State: enabled, Index: 1048581, Scope Policy: 5, Sequence
number: 1
  Source addresses:
    N/A: 0.0.0.0/2
  Destination addresses:
    N/A: 226.1.1.100/32
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: trust, To zone: untrust
  Policy: scope1-0002, State: enabled, Index: 2097156, Scope Policy: 4, Sequence
number: 2
```


Copyright © 2011, Juniper Networks, Inc.

```
  Source addresses:
    N/A: 0.0.0.0/2
  Destination addresses:
    N/A: 0.0.0.0/2
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
From zone: untrust, To zone: trust
  Policy: scope1-0002, State: enabled, Index: 2097157, Scope Policy: 5, Sequence
number: 2
  Source addresses:
    N/A: 0.0.0.0/2
  Destination addresses:
    N/A: 0.0.0.0/2
    Applications: Unknown([0-0]->[0-0]/0)
    Action: permit, tunnel
```

Multicast Encrypting Side Data Path

1. A UDP packet is sent from 31.1.1.2 to 226.1.1.100.
2. The packet reaches “black” on the trust zone.
3. A route look up is done for the multicast address 226.1.1.100. We then figure out that the outgoing interface for the packet to 226.1.1.100 is in the untrust zone. Note there can be more than one outgoing interface based on the multicast route lookup. In this example it is just one.
4. A policy lookup is done from the trust zone to the untrust zone and matches the dynamic policy that was pushed down from the key server.
5. Seven sessions are created.

```
Session ID: 36688, Policy name: N/A, Timeout: 1800, Valid

  In: 31.1.1.2/1 --> 226.1.1.100/1;61, If: ge-0/0/15.0, Pkts: 0, Bytes: 0
  Out: 255.255.255.255/1 --> 255.255.255.255/1;61, If: .local..0, Pkts: 0, Bytes:
0

Session ID: 36689, Policy name: scope1-0001/1048580, Timeout: -1, Valid
  In: 31.1.1.2/1 --> 226.1.1.100/1;61, If: ge-0/0/15.0, Pkts: 3848825, Bytes:
177045950
  Out: 226.1.1.100/1 --> 31.1.1.2/1;61, If: ge-0/0/11.0, Pkts: 0, Bytes: 0

Session ID: 36690, Policy name: scope1-0001/1048580, Timeout: -1, Valid
  In: 31.1.1.2/1 --> 226.1.1.100/1;61, If: ge-0/0/15.0, Pkts: 3848836, Bytes:
177046456
  Out: 226.1.1.100/1 --> 31.1.1.2/1;61, If: ge-0/0/10.0, Pkts: 0, Bytes: 0

Session ID: 36691, Policy name: N/A, Timeout: N/A, Allocated
  In: 226.1.1.100/374 --> 31.1.1.2/65070;esp, If: ge-0/0/10.0, Pkts: 0, Bytes: 0

Session ID: 36692, Policy name: N/A, Timeout: N/A, Allocated
  In: 226.1.1.100/374 --> 31.1.1.2/65070;esp, If: ge-0/0/11.0, Pkts: 0, Bytes: 0

Session ID: 36693, Policy name: N/A, Timeout: N/A, Allocated
  In: 226.1.1.100/0 --> 31.1.1.2/0;esp, If: N/A, Pkts: 0, Bytes: 0

Session ID: 36694, Policy name: N/A, Timeout: N/A, Allocated
  In: 226.1.1.100/0 --> 31.1.1.2/0;esp, If: N/A, Pkts: 0, Bytes: 0
```

Copyright © 2011, Juniper Networks, Inc.

6. Session “36688” is a template session. At this point it is not known how many fanouts are going to be there.
7. Now the scope policy is matched and two plain-text sessions are created. There are two sessions, as the route to 226.1.1.100 is in inet.1—and two copies have to be sent 226.1.1.100, as there are two different multicast receivers, each reachable via different interfaces. Sessions “36689” and “36690” correspond to this.
8. Tunnel sessions are created, one for each stream to 226.1.1.2. Sessions “36691” and “36692” correspond to this. The packet is encrypted and then forwarded to the destinations based on routes obtained via PIM. It is important to note that the multicast fanout occurs only after encryption.
9. Like the unicast case, two shadow tunnel sessions are created—one for each stream. Sessions “36693” and “36694” correspond to this.

Multicast Decrypting Side Data Path

1. For the UDP from 31.1.1.2 to 226.1.1.1.100, the encrypted packet arrives on the untrust interface (ge-0/0/10.0) on “white.” It also arrives to “rainbow.” The data path is exactly the same as white—only the white case is explained in this section.
2. This packet matches the 0.0.0.0/xxxxx --> 0.0.0.0/xxxxx session created for the IPsec SA/SPI after the key server’s pushed down policy.
3. Four new sessions similar to the encrypt side are created.

```
Session ID: 27624, Policy name: N/A, Timeout: N/A, Valid
  In: 31.1.1.2/18885 --> 226.1.1.100/60773;esp, If: lo0.0, Pkts: 0, Bytes: 0

Session ID: 27625, Policy name: N/A, Timeout: N/A, Valid
  In: 31.1.1.2/0 --> 226.1.1.100/0;esp, If: lo0.0, Pkts: 0, Bytes: 0

Session ID: 32170, Policy name: N/A, Timeout: 1800, Valid
  In: 31.1.1.2/1 --> 226.1.1.100/1;61, If: ge-0/0/10.0, Pkts: 0, Bytes: 0
  Out: 255.255.255.255/1 --> 255.255.255.255/1;61, If: .local..0, Pkts: 0, Bytes:
0

Session ID: 32171, Policy name: scope1-0004/4194309, Timeout: -1, Valid
  In: 31.1.1.2/1 --> 226.1.1.100/1;61, If: ge-0/0/10.0, Pkts: 49223414, Bytes:
2264277044
  Out: 226.1.1.100/1 --> 31.1.1.2/1;61, If: ge-0/0/1.0, Pkts: 0, Bytes: 0
```

4. Session “27624” is the tunnel session, and session “27625” is the shadow tunnel session.
5. Sessions “32170” and “32171” are created for the multicast plain-text traffic.
6. The packet is then decrypted and sent to the multicast destination of 41.1.1.2.

Group VPN Advanced Knobs

There might be scenarios where the key server seeks to be a member of another group. You can define both member and server configurations on a single SRX Series chassis. To enable this cooperation, you need to define the member and server configurations under the “group-vpn colocation hierarchy.”

```
root@data-center# set security group-vpn co-location ?
Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don’t inherit configuration data from these groups
> member               Group VPN member configuration
> server               Group VPN server configuration
  |                    Pipe through a command
[edit]
```


Copyright © 2011, Juniper Networks, Inc.

Group VPN provides anti-replay protection like traditional VPN. However, anti-replay in the Group VPN protocol is based on time (pseudo-time) and not sequence number like traditional VPNs. The key server is responsible for establishing and maintaining the pseudo-time for a group. It must also keep pseudo-time synchronized on all group members via rekey updates. Every group member includes its pseudo-time as a timestamp in the data packets. A receiving VPN gateway then compares the timestamp of the received packet with the group member reference pseudo-time clock it maintains for the group. If the packet arrives too late, it is dropped. You can define the time window to specify whether or not a packet arrived late by using the following configuration:

```
set security group-vpn server group g1 anti-replay-time-window 60
```

The anti-replay time window can be between 60 and 360 seconds. Anti-replay is enabled by default and can be disabled with the config knob:

```
set security group-vpn server group g1 no-anti-replay
```

You can also configure a delay for activating a group-vpn using the config knob:

```
set security group-vpn server group g1 activation-time-delay 30
```

Group VPN Redundancy

Currently, Juniper's Group VPN solution does not support high availability or cooperative key servers. Our solution to redundancy, both at the key server as well as the group member, is to use VRRP.

Key Server VRRP

For key server redundancy, you need two key servers, each having the same configuration for the group VPN part. For the `server-address` definition under the group VPN server configuration, you must provide the VRRP IP address and not the IP address of the individual interface. You also need to additionally configure VRRP on both members. This is pretty much all you need to enable server redundancy via VRRP.

The following is an example of the vrrp configuration on the interfaces of both servers. The interfaces have individual addresses of 1.1.1.2/24 and 1.1.1.3/24 and share an IP address of 1.1.1.1/24. All communication to 1.1.1.1 is received by one of the two servers, the one which is the primary. The process of which server is elected as primary is beyond the scope of this document. In the Group VPN group members, the IP address defined for the server is 1.1.1.1.

```
Server1 :

fe-0/0/7 {
    unit 0 {
        family inet {
            address 1.1.1.2/24 {
                vrrp-group 1 {
                    virtual-address 1.1.1.1;
                    accept-data;
                }
            }
        }
    }
}

Server2 :

fe-0/0/6 {
    unit 0 {
        family inet {
```

Copyright © 2011, Juniper Networks, Inc.

```
            address 1.1.1.3/24 {
                vrrp-group 1 {
                    virtual-address 1.1.1.1;
                    accept-data;
                }
     }
    }
  }
}
```

Key Server VRRP limitations

1. The only server member communication supported with the VRRP solution is “Unicast PULL.” Both unicast and multicast push are not supported.
2. There is a case with the VRRP solution where some group members cannot communicate with other group members. Consider two key servers in the VRRP solution, key servers X and Y. Assume that there are three members—A, B, and C. Via VRRP election, X is the key server and all three members get their Phase 2 group keys from server X. Assume that the lifetime of the key is 1800 seconds. Assume that after 600 seconds key server X dies—and now by VRRP, key server Y takes over. Even at this point you are fine. All the members can communicate with each other via keys obtained from key server X. Now after 1000 seconds assume that member A reboots. Once it is back up, member A obtains its group key from key server Y. This key is different from the key members that B and C have. And hence, A cannot communicate with B or C. The key to keep this “out-of-sync” time to a minimum is to have a small rekey timer.

Group Member VRRP

For group member redundancy, you can have two group members with exactly the same configuration, each having unique IP addresses, but sharing a unique VRRP IP address for the interface that is used to communicate with the key server. It is imperative that you make sure that traffic to be encrypted enters the group member, which is the active VRRP member for the VRRP interface that is used to communicate with the key server. To ensure this, it is desired that you have another VRRP interface through which traffic enters the group member. Both VRRP groups must be active on the same SRX Series chassis.

It is important that the member address that you define on the key server is the VRRP address of the group member and not its individual IP addresses. This prevents both members from obtaining group SA from the key server.

```
security group-vpn server ike gateway rainbow address vrrp-ip-address
```

Group Member VRRP Limitations

1. The secondary node of the VRRP group tries to unsuccessfully retrieve (using its own IP) the group SA. You cannot avoid this unnecessary communication. This can be a problem when you have a large number of group members.
2. There is not stateful synchronization of SA data between the two nodes of the VRRP group. Hence, if one node dies, there is loss of data until the new node does a PULL and gets the SA data from the key server.

Copyright © 2011, Juniper Networks, Inc.

Group VPN Interoperability

SRX Series devices can interoperate with a Cisco ISR for Group VPN as per the following table.

Table 1: INTEROP

MEMBER	SERVER	REKEY MECHANISM	ANTI-REPLAY PROTECTION	INTEROP
SRX Series	CISCO ISR	PULL	NO	YES
SRX Series	CISCO ISR	PULL	YES	NO
SRX Series	CISCO ISR	PUSH	NO/YES	NO
CISCO ISR	SRX Series	PULL/PUSH	NO/YES	NO

Group VPN Performance and Scaling

Group VPN performance on each platform is exactly the same as standard IPsec/VPN on the platform. Group VPN scaling is described in the following table.

Table 2. Group VPN Scaling

PLATFORM	MAX GM PER GROUP	MAX GROUPS	MAX SA PER GROUP	MAX POLICIES PER SA
SRX100	50	5	150	100
SRX210	75	10	200	150
SRX240	300	50	250	200
SRX650	1500	250	400	300

Group VPN Debugging

The basic steps to debug Group VPN are:

1. Verify if IKE SAs are established on key server and member.
2. Verify if IPsec SAs are up on key server and member.
3. Verify if KEK SA is established.
4. Enable trace-options on both member and key server if any of the SAs are not established in steps 1 through 3.
   a. Use the following config on both member and server: “set security ipsec traceoptions flag all,” “set security traceoptions flag all.”
   b. View member traceoptions at “/var/log/kmd” and key server traceoptions at “/var/log/gksd.”
   c. View logs to see why the SAs are not up.
5. Verify if dynamic policies are pushed from key server to member.
6. Verify that each member has a tunnel session for each SA.
7. Pass traffic and verify session and flow logs on each member.

Group VPN Limitations

Redundancy

Juniper’s Group VPN solution as of Junos OS Release 10.2 is not supported in HA. Also, there is no support for cooperative key servers wherein two key servers can maintain group membership state between them and members can simultaneously register to both key servers. In this solution a key server can take over if the other key server dies.

Private Address Space

The Group VPN solution requires globally routable addresses even for hosts behind a VPN gateway. Hence, the Group VPN solution does not work over the Internet or in NAT environments.

Copyright © 2011, Juniper Networks, Inc.

Group VPN Versus Other Solutions

Table 3. VPN Solution Comparisons

	GROUP VPN	DYNAMIC MULTIPOINT VPN	AUTO CONNECT VPN
OS Support	Junos OS and IOS	IOS	ScreenOS
Phase 2 SA	One	Multiple	Multiple
Work over Internet	No	Yes	Yes
Outer IP Header	Endpoint addresses	Gateway addresses	Gateway addresses
Redundancy	Not available in Junos OS	Yes	Yes
Underlying Technology	GDOI	GRE	NHRP
Anti-replay mechanism	Time-based	Sequence-based	Sequence-based

About Juniper Networks

Juniper Networks is in the business of network innovation. From devices to data centers, from consumers to cloud providers, Juniper Networks delivers the software, silicon and systems that transform the experience and economics of networking. The company serves customers and partners worldwide. Additional information can be found at **www.juniper.net**.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

APAC Headquarters
Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters
Juniper Networks Ireland
Airside Business Park
Swords, County Dublin, Ireland
Phone: 35.31.8903.600
EMEA Sales: 00800.4586.4737
Fax: 35.31.8903.601

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

Copyright 2011 Juniper Networks, Inc. All rights reserved. Juniper Networks, the Juniper Networks logo, Junos, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, 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.

3500202-001-EN Feb 2011 Printed on recycled paper

Copyright © 2011, Juniper Networks, Inc.