DMVPN Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes how to configure DMVPN on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

DMVPN Overview

This chapter gives an overview of Dynamic Multipoint Virtual Private Network (DMVPN) support on the AT&T Vyatta vRouter.

Simplification of hub-and-spoke topologies

Many large IP Security (IPsec) virtual private networks (VPNs) use a hub-and-spoke topology to reduce the number of connections required for full connectivity. But even a hub-and-spoke IPsec VPN network can be difficult to scale for any of the following reasons:

- Hub configuration can become exceedingly complex when there are many spoke devices because VPN endpoints are statically configured. This problem is exacerbated in networks when addressing is frequently changed.
- A full set of tunnels consumes a great many IP addresses because every set of tunnel endpoints requires a separate IP address space.
- The hub becomes a single point of failure for the network.
- The hub must process all network traffic and can become a processing bottleneck.

A dynamic multipoint VPN improves scaling for hub-and-spoke networks by allowing IPsec tunnels to be dynamically added as needed, without configuration. This greatly simplifies hub configuration and reduces the need for IP address space. In addition, after the hub-and-spoke network has been dynamically built out, network spokes can learn to communicate directly with each other thereby reducing the burden on the hub.

Components of DMVPN

DMVPN employs the following components:

- mGRE
- NHRP
- IPsec
- Routing Protocol

MGRE

The Generic Routing Encapsulation (GRE) protocol provides a simple general-purpose mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. In DMVPN, GRE encapsulates IP packets and transports them over VPN tunnels. An example is multicast routing advertisements, which are multicast. IPsec, which is a standard mechanism for providing security on IP networks, cannot encrypt multicast packets. However, multicast packets can be encapsulated within a GRE tunnel and then routed over a VPN connection, so that the encapsulated packets are protected by the IPsec tunnel.

Multipoint GRE (mGRE) allows an interface to support multiple GRE tunnels. In a DMVPN, multipoint mGRE tunnels are used to establish and aggregate the tunnels from the spokes to the hub.

mGRE commands are described in AT&T Vyatta Network Operating System Tunnels Configuration Guide.

NHRP

To build the dynamic tunnels, mGRE uses the Next Hop Resolution Protocol (NHRP) addressing service. The hub router maintains an NHRP database, acting as a route server. Spoke routers register their public IP addresses with the hub, acting as clients. The spokes query the hub database to obtain the IP addresses of the logical tunnel endpoints.

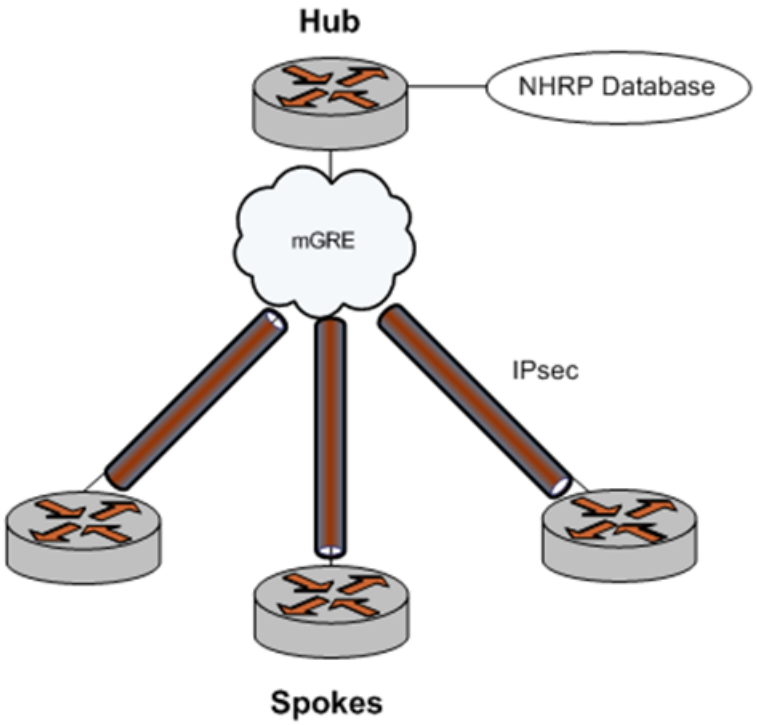
NHRP commands are described in AT&T Vyatta Network Operating System Services Configuration Guide.

IPsec

In DMVPN, tunnels are secured using the IP Security (IPsec). IPsec is a suite of protocols that protect network communication at the IP level (Layer 3).

Note: The AT&T Vyatta vRouter supports the AES options with 128-bit or 256-bit Galois/Counter Mode (GCM), which provides improved efficiency and performance.

Figure 1: IPsec



Routing protocol

DMVPN uses a dynamic routing protocol to advertise the private networks within the DMVPN network. The AT&T Vyatta vRouter supports the Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and the Border Gateway Protocol (BGP).

DMVPN in a spoke-to-spoke network

The DMVPN network is a hub-and-spoke network as the hub discovers all spokes on the network. The discovery process proceeds as follows:

1. The spoke must be configured with the address of the hub, which should be static.
2. Each spoke establishes a permanent IPsec tunnel to the hub.
3. The spoke registers with the hub, an NHRP Next Hop Server (NHS), as a Next Hop Client (NHC).
4. The spoke provides the hub with its real IP address.
5. The hub adds the spoke to its learned network (the NHRP database), mapping the real public IP address onto the logical VPN address for the spoke.

   Info:

   After the hub-and-spoke network has been built out, it can convert to a spoke-to-spoke network, as follows:

a. When a spoke has to communicate with a second spoke, it sends an NHRP query to the hub by using the logical VPN address for the second device.
b. The hub consults its NHRP database and replies with the real IP address of the second spoke.
c. Using the real IP address, the first spoke can dynamically set up an IPsec tunnel directly to the other spoke.
d. The tunnel is created on demand and bypasses the hub.

Restrictions

If you use the Routing Information Protocol (RIP) as the routing protocol in a hub-and-spoke network, you must disable split horizon. Split horizon prevents spokes from receiving advertisements about other spokes. For more information about split horizon, see the **`interfaces`** *<interface>* **`ip rip split-horizon`** command in AT&T Vyatta Network Operating System RIP Configuration Guide.

Support of x509 authentication for DMVPN

The following commands are used to support x509 authentication for DMVPN.

For more information, refer to AT&T Vyatta Network Operating System IPsec Site-to-Site VPN Configuration Guide.

- set security vpn ipsec profile DMVPN authentication mode
- set security vpn ipsec profile DMVPN authentication remote-id
- set security vpn ipsec profile DMVPN authentication x509 cert-file
- set security vpn ipsec profile DMVPN authentication x509 key file
- set security vpn ipsec profile DMVPN authentication x509 key password

Supported standards

The AT&T Vyatta vRouter implementation of GRE complies with the following standards:

- RFC 1702: Generic Routing Encapsulation over IPv4 Networks
- RFC 2784: Generic Routing Encapsulation

The AT&T Vyatta vRouter implementation of NHRP complies with the following standard:

- RFC 2332: NBMA Next Hop Resolution Protocol (NHRP)

The AT&T Vyatta vRouter implementation of IPsec complies with the following standards:

- RFC 2401, Security Architecture for the Internet Protocol
- RFC 2402, IP Authentication Header (AH)
- RFC 2406, IP Encapsulating Security Payload (ESP)
- RFC 2407, The Internet IP Security Domain of Interpretation for ISAKMP
- RFC 2408, Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 2409, The Internet Key Exchange (IKE)
- RFC 2412, The OAKLEY Key Determination Protocol
- RFC 3526, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
- RFC 3706, A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
- RFC 4307, Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
- RFC 4478, Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
- RFC 7296, Internet Key Exchange Protocol Version 2 (IKEv2)
- RFC 7815, Minimal Internet Key Exchange Version 2 (IKEv2) Initiator Implementation

Related guides

In addition to the information presented in this guide, you can refer to the following documents:

- AT&T Vyatta Network Operating System Tunnels Configuration Guide presents information on the Generic Routing Encapsulation (GRE), including multipoint GRE (mGRE), which is a component of DMVPN. GRE and mGRE commands are described in that guide.
- AT&T Vyatta Network Operating System Services Configuration Guide presents information on Next Hop Resolution Protocol (NHRP), which is a component of DMVPN. NHRP commands are described in that guide.
- AT&T Vyatta Network Operating System IPsec Site-to-Site VPN Configuration Guide presents additional information about vRouter support for the IP Security (IPsec) suite of protocols. IPsec commands are described in that guide.
- AT&T Vyatta Network Operating System RIP Configuration Guide, AT&T Vyatta Network Operating System RIPng Configuration Guide, AT&T Vyatta Network Operating System OSPF Configuration Guide, and AT&T Vyatta Network Operating System BGP Configuration Guide present information about the dynamic routing protocols supported by the AT&T Vyatta vRouter.

DMVPN Configuration Examples

This chapter provides multipoint Generic Routing Encapsulation (GRE) and Dynamic Multipoint Virtual Private Network (DMVPN) configuration examples.

Note: By default, rekeying is enabled for a DMVPN hub, which allows the hub to renegotiate a connection when it is about to expire.

Prerequisites

The examples in this chapter have some elements in common:

- Any Ethernet interface to be used must already be configured. The examples do not show Ethernet interface configurations.
- Loopback or Ethernet interfaces are typically configured as tunnel endpoints. Configuring a loopback interface as the tunnel endpoint is advantageous in systems in which there are multiple paths between tunnel endpoints. If the endpoint is the loopback interface, the tunnel does not fail if an Ethernet interface fails.

See AT&T Vyatta Network Operating System LAN Interfaces Configuration Guide for information about configuring Ethernet and loopback interfaces.

Basic multipoint GRE tunnel

This section presents a sample configuration for basic multipoint Generic Routing Encapsulation (mGRE) tunnels between AT&T Vyatta vRouter HUB and SPOKE1, and HUB and SPOKE2. The configuration shown in this example also provides for a dynamic tunnel to be created between SPOKE1 and SPOKE2 as required. This ability derives from the use of multipoint GRE and NHRP. This configuration can be expanded by creating additional spoke nodes with no change to the HUB configuration.

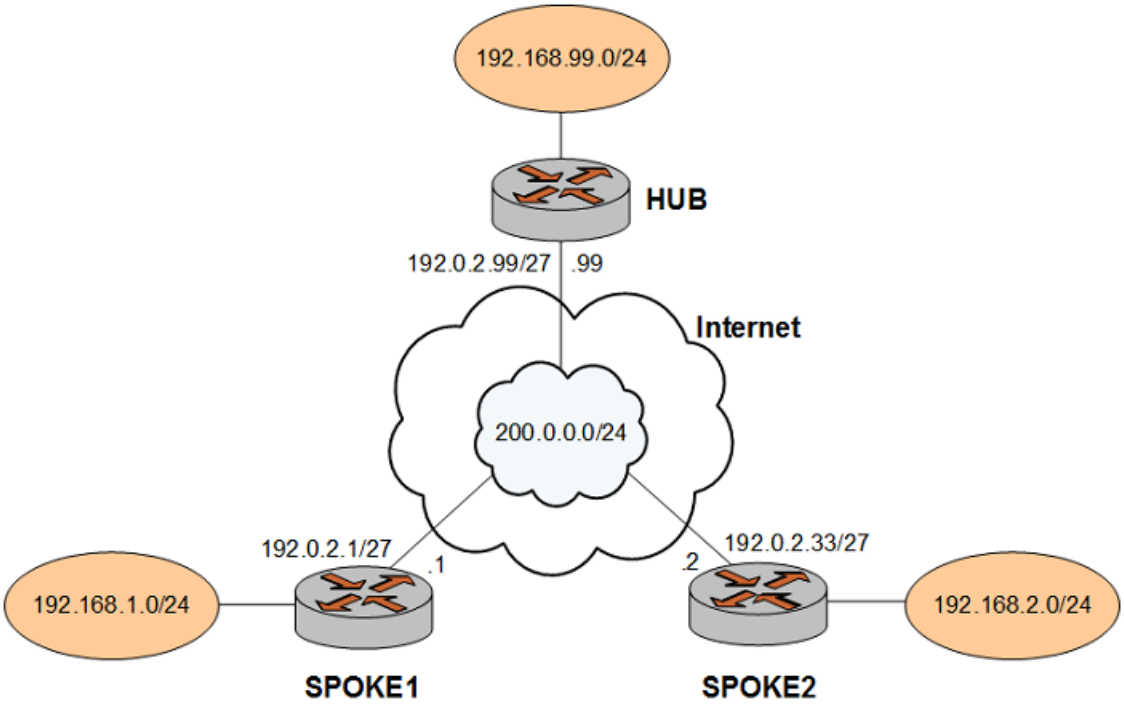
- For more information on mGRE, including mGRE commands, see AT&T Vyatta Network Operating System Tunnels Configuration Guide.
- For more information on NHRP, including NHRP commands, see AT&T Vyatta Network Operating System Services Configuration Guide.

Note that spoke-to-spoke traffic does not pass through the HUB router. Also note that a typical production environment would use a routing protocol such as OSPF rather than using the static routes that are used in the example.

The basic mGRE tunnels presented in this example are not protected by IPsec encryption, which means they are not secure and would not be suitable for a production network unless otherwise secured. DMVPN uses mGRE, NHRP, and IPsec to provide a secure hub-and-spoke tunnel environment. For an example of a full DMVPN configuration, see the following section DMVPN hub-and-spoke *(page 15)*.

When this example is completed, the network will be configured as shown in the following figure.

Figure 2: Basic mGRE tunnel network



Configure HUB

Two multipoint GRE tunnels are configured. One is between HUB and SPOKE1. The other is between HUB and SPOKE2. The first step is to configure HUB.

In this example, you create the tunnel interface and the tunnel endpoint on HUB.

- The tunnel interface tun0 on HUB is assigned the IP address 200.0.0.99 on subnet 200.0.0.0/24.
- The source IP address of the tunnel endpoint (the `local-ip`) is the same as the address associated with the local Ethernet interface in this example (192.0.2.99/24).
- A static route is created to specify how to get to the remote LANs through the tunnel.

Table 1: Creating a multipoint GRE endpoint on HUB

Step	Command
Create the tunnel interface, and specify the IP address to be associated with it.	`vyatta@HUB# set interfaces tunnel tun0 address 200.0.0.99/24`
Specify the encapsulation mode for the tunnel.	`vyatta@HUB# set interfaces tunnel tun0 encapsulation gre-multipoint`
Specify the source IP address for the tunnel. This address is the IP address of the physical interface for the tunnel endpoint.	`vyatta@HUB# set interfaces tunnel tun0 local-ip 192.0.2.99`
Specify that Cisco-style NHRP Traffic Indication packets are to be sent.	`vyatta@HUB# set interfaces tunnel tun0 nhrp redirect`

Step	Command
Commit the configuration.	`vyatta@HUB# commit`

View the configuration.	`vyatta@HUB# show interfaces tunnel`
	
	`tun0 {`
	`  address 200.0.0.99/24`
	`  encapsulation gre-multipoint`
	`  local-ip 192.0.2.99`
	`  nhrp {`
	`  redirect`
	`  }`
	`}`

Create a static route to access the remote LAN behind SPOKE1 through the tunnel.	`vyatta@HUB# set protocols static route 192.168.1.0/24 next-hop 200.0.0.1`
Create a static route to access the remote LAN behind SPOKE2 through the tunnel.	`vyatta@HUB# set protocols static route 192.168.2.0/24 next-hop 200.0.0.2`
Commit the configuration.	`vyatta@HUB# commit`

View the configuration.	`vyatta@HUB# show protocols`
	`static {`
	`  route 192.168.1.0/24 {`
	`  next-hop 200.0.0.1 {`
	`  }`
	`  }`
	`  route 192.168.2.0/24 {`
	`  next-hop 200.0.0.2 {`
	`  }`
	`  }`
	`}`

Configure SPOKE1

The second step is to configure SPOKE1.

In this example, you create the tunnel interface and the tunnel endpoint on SPOKE1.

- The tunnel interface tun0 on HUB is assigned the IP address 200.0.0.1 on subnet 200.0.0.0/24.
- The source IP address of the tunnel endpoint (the `local-ip`) is the same as the address associated with the local Ethernet interface in this example (192.0.2.1/24).
- A static route is created to specify how to get to the remote LANs through the tunnel.

Table 2: Creating a multipoint GRE endpoint on SPOKE1

Step	Command
Create the tunnel interface, and specify the IP address to be associated with it.	`vyatta@SPOKE1# set interfaces tunnel tun0 address 200.0.0.1/24`

Step	Command
Specify the encapsulation mode for the tunnel.	`vyatta@SPOKE1# set interfaces tunnel tun0 encapsulation gre-multipoint`
Specify the source IP address for the tunnel. This address is the IP address of the physical interface for the tunnel endpoint.	`vyatta@SPOKE1# set interfaces tunnel tun0 local-ip 192.0.2.1`
Prevent multicast protocols (for example, routing protocols) being carried over the tunnel.	`vyatta@SPOKE1# set interfaces tunnel tun0 multicast disable`
Map the IP address of the tunnel interface of the Hub to its physical IP address.	`vyatta@SPOKE1# set interfaces tunnel tun0 nhrp map 200.0.0.99/24 nbma-address 192.0.2.99`
Specify that this spoke should register itself automatically on startup.	`vyatta@SPOKE1# set interfaces tunnel tun0 nhrp map 200.0.0.99/24 register`
Specify that Cisco-style NHRP Traffic Indication packets are to be sent.	`vyatta@SPOKE1# set interfaces tunnel tun0 nhrp redirect`
Specify that shortcut routes can be created.	`vyatta@SPOKE1# set interfaces tunnel tun0 nhrp shortcut`
Commit the configuration.	`vyatta@SPOKE1# commit`

View the configuration.	`vyatta@SPOKE1# show interfaces tunnel`
	
	`tun0 {`
	`  address 200.0.0.1/24`
	`  encapsulation gre-multipoint`
	`  local-ip 192.0.2.1`
	`  multicast disable`
	`  nhrp {`
	`  map 200.0.0.99/24 {`
	`  nbma-address 192.0.2.99`
	`  register`
	`  }`
	`  redirect`
	`  shortcut`
	`  }`
	`}`

Create a static route to access the remote LAN behind HUB through the tunnel.	`vyatta@HUB# set protocols static route 192.168.99.0/24 next-hop 200.0.0.99`
Create a static route to access the remote LAN behind SPOKE2 through the tunnel.	`vyatta@HUB# set protocols static route 192.168.2.0/24 next-hop 200.0.0.2`
Commit the configuration.	`vyatta@HUB# commit`

Step	Command

View the configuration.	`vyatta@HUB# show protocols`
	
	`static {`
	`route 192.168.99.0/24 {`
	`next-hop 200.0.0.99 {`
	`}`
	`}`
	`route 192.168.2.0/24 {`
	`next-hop 200.0.0.2 {`
	`}`
	`}`
	`}`

Configure SPOKE2

The final step is to configure SPOKE2.

In this example, you create the tunnel interface and the tunnel endpoint on SPOKE2.

- The tunnel interface tun0 on HUB is assigned the IP address 200.0.0.2 on subnet 200.0.0.0/24.
- The source IP address of the tunnel endpoint (the `local-ip`) is the same as the address associated with the local Ethernet interface in this example (192.0.2.33/24).
- A static route is created to specify how to get to the remote LANs through the tunnel

Table 3: Creating a multipoint GRE endpoint on SPOKE2

Step	Command
Create the tunnel interface, and specify the IP address to be associated with it.	`vyatta@SPOKE2# set interfaces tunnel tun0 address 200.0.0.2/24`
Specify the encapsulation mode for the tunnel.	`vyatta@SPOKE2# set interfaces tunnel tun0 encapsulation gre-multipoint`
Specify the source IP address for the tunnel. This address is the IP address of the physical interface for the tunnel endpoint.	`vyatta@SPOKE2# set interfaces tunnel tun0 local-ip 192.0.2.33`
Prevent multicast protocols (for example, routing protocols) being carried over the tunnel.	`vyatta@SPOKE2# set interfaces tunnel tun0 multicast disable`
Map the IP address of the tunnel interface of the Hub to its physical IP address.	`vyatta@SPOKE2# set interfaces tunnel tun0 nhrp map 200.0.0.99/24 nbma-address 192.0.2.99`
Specify that this spoke should register itself automatically on startup.	`vyatta@SPOKE2# set interfaces tunnel tun0 nhrp map 200.0.0.99/24 register`
Specify that Cisco-style NHRP Traffic Indication packets are to be sent.	`vyatta@SPOKE2# set interfaces tunnel tun0 nhrp redirect`

Step	Command
Specify that shortcut routes can be created.	`vyatta@SPOKE2# set interfaces tunnel tun0 nhrp shortcut`
Commit the configuration.	`vyatta@SPOKE2# commit`

View the configuration.	`vyatta@SPOKE2# show interfaces tunnel`
	`tun0 {`
	`  address 200.0.0.2/24`
	`  encapsulation gre-multipoint`
	`  local-ip 192.0.2.33`
	`  multicast disable`
	`  nhrp {`
	`  map 200.0.0.99/24 {`
	`  nbma-address 192.0.2.99`
	`  register`
	`  }`
	`  redirect`
	`  shortcut`
	`  }`
	`}`

Create a static route to access the remote LAN behind HUB through the tunnel.	`vyatta@HUB# set protocols static route 192.168.99.0/24 next-hop 200.0.0.99`
Create a static route to access the remote LAN behind SPOKE1 through the tunnel.	`vyatta@HUB# set protocols static route 192.168.1.0/24 next-hop 200.0.0.1`
Commit the configuration.	`vyatta@HUB# commit`

View the configuration.	`vyatta@HUB# show protocols`
	
	`static {`
	`  route 192.168.99.0/24 {`
	`  next-hop 200.0.0.99 {`
	`  }`
	`  }`
	`  route 192.168.1.0/24 {`
	`  next-hop 200.0.0.1 {`
	`  }`
	`  }`
	`}`

DMVPN hub-and-spoke

The basic mGRE tunnel environment presented in the previous example is not protected by IPsec encryption, which means they are not secure and would not be suitable for a production network unless otherwise secured. DMVPN uses mGRE, NHRP, and IPsec to provide a secure hub-and-spoke tunnel environment.

The previous example shows the mGRE and NHRP configuration. This section presents the IPsec configuration required to secure the environment shown in the previous example and provide a complete DMVPN solution. For more information on configuring IPsec site-to-site environments, see AT&T Vyatta Network Operating System IPsec Site-to-Site VPN Configuration Guide.

Configure HUB

This section describes how to configure HUB.

Configuring an IKE group on HUB

To create an Internet Key Exchange (IKE) group, perform the following steps on HUB in configuration mode.

Table 4: Configuring an IKE group on HUB

Step	Command
Create the configuration node for proposal 1 of IKE group IKE-1H.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H proposal 1`
(Optional) Specify version 2 of IKE (IKEv2).	`vyatta@HUB# set security vpn ipsec ike-group IKE-1W ike-version 2`
Set the encryption cipher for proposal 1.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H proposal 1 encryption aes256`
Set the hash algorithm for proposal 1.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H proposal 1 hash sha1`
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of IKE group IKE-1H.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H proposal 2 encryption aes128`
Set the hash algorithm for proposal 2.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H proposal 2 hash sha1`
Set the lifetime for the whole IKE group.	`vyatta@HUB# set security vpn ipsec ike-group IKE-1H lifetime 3600`

View the configuration for the IKE group. Do not commit yet.	`vyatta@HUB# show vpn ipsec ike-group IKE-1H`
	
	`>  proposal 1 {`
	`>  encryption aes256`
	`>  hash sha1`
	`>  }`
	`>  proposal 2 {`
	`>  encryption aes128`
	`>  hash sha1`
	`>  }`
	`>  lifetime 3600`

Configuring an ESP group on HUB

To create an ESP group, perform the following steps on HUB in configuration mode.

Table 5: Configuring an ESP group on HUB

Step	Command
Create the configuration node for proposal 1 of ESP group ESP-1H.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H proposal 1`
Set the encryption cipher for proposal 1.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H proposal 1 encryption aes256`
Set the hash algorithm for proposal 1.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H proposal 1 hash sha1`
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of ESP group ESP-1H.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H proposal 2 encryption aes128gcm128`
Set the hash algorithm for proposal 2.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H proposal 2 hash null`
Set the lifetime for the whole ESP group.	`vyatta@HUB# set security vpn ipsec esp-group ESP-1H lifetime 1800`

View the configuration for the ESP group. Do not commit yet.	`vyatta@HUB# show vpn ipsec esp-group ESP-1H`
	`>  proposal 1 {`
	`>  encryption aes256`
	`>  hash sha1`
	`>  }`
	`>  proposal 2 {`
	`>  encryption aes128gcm128`
	`>  hash null`
	`>  }`
	`>  lifetime 1800`

Securing the mGRE tunnel on HUB

To secure the mGRE tunnel with IPsec, perform the following steps on HUB in configuration mode.

Table 6: Securing the mGRE tunnel with IPsec

Step	Command
Create the profile node.	`vyatta@HUB# set security vpn ipsec profile DMVPN`
Set the authentication mode.	`vyatta@HUB# set security vpn ipsec profile DMVPN authentication mode pre-shared-secret`
Define the preshared secret key. It must match that set on remote systems.	`vyatta@HUB# set security vpn ipsec profile DMVPN authentication pre-shared-secret NET123`

Step	Command
Bind the IPsec configuration to the tunnel.	`vyatta@HUB# set security vpn ipsec profile DMVPN bind tunnel tun0`
Specify the ESP configuration to use.	`vyatta@HUB# set security vpn ipsec profile DMVPN esp-group ESP-1H`
Specify the IKE configuration to use.	`vyatta@HUB# set security vpn ipsec profile DMVPN ike-group IKE-1H`
Commit the configuration.	`vyatta@HUB# commit`

View the configuration for the profile.	`vyatta@HUB# show vpn ipsec profile DMVPN`
	
	`authentication {`
	`  mode pre-shared-secret`
	`  pre-shared-secret NET123`
	`}`
	`bind {`
	`  tunnel tun0`
	`}`
	`esp-group ESP-1H`
	`ike-group IKE-1H`

Configure SPOKE1

This section describes how to configure SPOKE1.

Configuring an IKE group on SPOKE1

To create an IKE group, perform the following steps on SPOKE1 in configuration mode.

Table 7: Configuring an IKE group on SPOKE1

Step	Command
Create the configuration node for proposal 1 of IKE group IKE-1S.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S proposal 1`
(Optional) Specify version 2 of IKE (IKEv2).	`vyatta@HUB# set security vpn ipsec ike-group IKE-1W ike-version 2`
Set the encryption cipher for proposal 1.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S proposal 1 encryption aes256`
Set the hash algorithm for proposal 1.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S proposal 1 hash sha1`
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of IKE group IKE-1S.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S proposal 2 encryption aes128`

Step	Command
Set the hash algorithm for proposal 2.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S proposal 2 hash sha1`
Set the lifetime for the whole IKE group.	`vyatta@SPOKE1# set security vpn ipsec ike-group IKE-1S lifetime 3600`

View the configuration for the IKE group. Do not commit yet.	`vyatta@SPOKE1# show vpn ipsec ike-group IKE-1S`
	`>  proposal 1 {`
	`>  encryption aes256`
	`>  hash sha1`
	`>  }`
	`>  proposal 2 {`
	`>  encryption aes128`
	`>  hash sha1`
	`>  }`
	`>  lifetime 3600`

Configuring an ESP group on SPOKE1

To create an ESP group, perform the following steps on SPOKE1 in configuration mode.

Table 8: Configuring an ESP group on SPOKE1

Step	Command
Create the configuration node for proposal 1 of ESP group ESP-1S.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1S proposal 1`
Set the encryption cipher for proposal 1.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1S proposal 1 encryption aes256`
Set the hash algorithm for proposal 1.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1S proposal 1 hash sha1`
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of ESP group ESP-1S.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1H proposal 2 encryption aes128gcm128`
Set the hash algorithm for proposal 2.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1S proposal 2 hash null`
Set the lifetime for the whole ESP group.	`vyatta@SPOKE1# set security vpn ipsec esp-group ESP-1S lifetime 1800`

Step	Command

View the configuration for the ESP group. Do not commit yet.	`vyatta@SPOKE1# show vpn ipsec esp-group ESP-1S`
	
	`>  proposal 1 {`
	`>  encryption aes256`
	`>  hash sha1`
	`>  }`
	`>  proposal 2 {`
	`>  encryption aes128gcm128`
	`>  hash null`
	`>  }`
	`>  lifetime 1800`

Securing the mGRE tunnel with IPsec on SPOKE1

To secure the mGRE tunnel with IPsec, perform the following steps on SPOKE1 in configuration mode.

Table 9: Securing the mGRE tunnel with IPsec

Step	Command
Create the profile node.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN`
Set the authentication mode.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN authentication mode pre-shared-secret`
Define the preshared secret key. It must match that set on remote systems.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN authentication pre-shared-secret NET123`
Bind the IPsec configuration to the tunnel.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN bind tunnel tun0`
Specify the ESP configuration to use.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN esp-group ESP-1S`
Specify the IKE configuration to use.	`vyatta@SPOKE1# set security vpn ipsec profile DMVPN ike-group IKE-1S`
Commit the configuration.	`vyatta@SPOKE1# commit`

Step	Command

View the configuration for the profile.	`vyatta@SPOKE1# show vpn ipsec profile DMVPN`
	
	`authentication {`
	`mode pre-shared-secret`
	`pre-shared-secret NET123`
	`}`
	`bind {`
	`tunnel tun0`
	`}`
	`esp-group ESP-1S`
	`ike-group IKE-1S`

Configure SPOKE2

This section describes how to configure SPOKE2.

Configuring an IKE group on SPOKE2

To create an IKE group, perform the following steps on SPOKE2 in configuration mode.

Table 10: Configuring an IKE group on SPOKE2

Step	Command
Create the configuration node for proposal 1 of IKE group IKE-1S.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S proposal 1`
(Optional) Specify version 2 of IKE (IKEv2).	`vyatta@HUB# set security vpn ipsec ike-group IKE-1W ike-version 2`
Set the encryption cipher for proposal 1.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S proposal 1 encryption aes256`
Set the hash algorithm for proposal 1.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S proposal 1 hash sha1`
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of IKE group IKE-1S.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S proposal 2 encryption aes128`
Set the hash algorithm for proposal 2.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S proposal 2 hash sha1`
Set the lifetime for the whole IKE group.	`vyatta@SPOKE2# set security vpn ipsec ike-group IKE-1S lifetime 3600`

Step	Command

View the configuration for the IKE group. Do not commit yet.	vyatta@SPOKE2# show vpn ipsec ike-group IKE-1S
	
	> proposal 1 {
	> encryption aes256
	> hash sha1
	> }
	> proposal 2 {
	> encryption aes128
	> hash sha1
	> }
	> lifetime 3600

Configuring an ESP group on SPOKE2

To create an ESP group, perform the following steps on SPOKE2 in configuration mode.

Table 11: Configuring an ESP group on SPOKE2

Step	Command
Create the configuration node for proposal 1 of ESP group ESP-1S.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S proposal 1
Set the encryption cipher for proposal 1.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S proposal 1 encryption aes256
Set the hash algorithm for proposal 1.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S proposal 1 hash sha1
Set the encryption cipher for proposal 2. This action also creates the configuration node for proposal 2 of ESP group ESP-1S.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S proposal 2 encryption aes128gcm128
Set the hash algorithm for proposal 2.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S proposal 2 hash null
Set the lifetime for the whole ESP group.	vyatta@SPOKE2# set security vpn ipsec esp-group ESP-1S lifetime 1800

View the configuration for the ESP group. Do not commit yet.	vyatta@SPOKE2# show vpn ipsec esp-group ESP-1S
	
	> proposal 1 {
	> encryption aes256
	> hash sha1
	> }
	> proposal 2 {
	> encryption aes128gcm128
	> hash null
	> }
	> lifetime 1800

Securing the mGRE tunnel on SPOKE2

To secure the mGRE tunnel with IPsec, perform the following steps on SPOKE2 in configuration mode.

Table 12: Securing the mGRE tunnel with IPsec

Step	Command
Create the profile node.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN`
Set the authentication mode.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN authentication mode pre-shared-secret`
Define the preshared secret key. It must match that set on remote systems.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN authentication pre-shared-secret NET123`
Bind the IPsec configuration to the tunnel.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN bind tunnel tun0`
Specify the ESP configuration to use.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN esp-group ESP-1S`
Specify the IKE configuration to use.	`vyatta@SPOKE2# set security vpn ipsec profile DMVPN ike-group IKE-1S`
Commit the configuration.	`vyatta@SPOKE2# commit`

View the configuration for the profile.	`vyatta@SPOKE2# show vpn ipsec profile DMVPN`
	`authentication {`
	`  mode pre-shared-secret`
	`  pre-shared-secret NET123`
	`}`
	`bind {`
	`  tunnel tun0`
	`}`
	`esp-group ESP-1S`
	`ike-group IKE-1S`

Deploying DMVPN for multiple end users from the same hub site

You can deploy DMVPN for multiple end users from the same hub site by segregating the user traffic into separate VRF routing instances. By assigning a GRE overlay tunnel interface to a routing instance, the tunnel interface becomes VRF-aware. All of the other VRF components, including IPsec control plane, IPsec data plane, and GRE underlay, are not VRF-aware. They remain in the default routing instance, and no change is required for their configuration.

The following example shows how to associate a tunnel interface with a routing instance. In the example, tunnel interface tun0 is assigned to routing instance vrf-1.

```
vyatta@vyatta# set routing routing-instance vrf-1 instance-type vrf
vyatta@vyatta# set routing routing-instance vrf-1 interface tun0
```

Use the following command to view the configuration. Notice that the routing instance is included in the output.

```
vyatta@vyatta# run show ip nhrp
Status: ok

Interface: tun0
Type: local
Protocol-Address: 40.40.40.255/32
Alias-Address: 40.40.40.1
Flags: up
Routing instance: vrf-1

Interface: tun0
Type: local
Protocol-Address: 40.40.40.1/32
Flags: up

Interface: tun0
Type: dynamic
Protocol-Address: 40.40.40.2/32
NBMA-Address: 33.0.0.2
Flags: up
Expires-In: 4:23
```

See AT&T Vyatta Network Operating System Basic Routing Configuration Guide for additional information about VRF and routing instances.

DMVPN Commands

This chapter directs you to guides that describe commands used to implement Dynamic Multipoint Virtual Private Network (DMVPN) configuration.

DMVPN commands

DMVPN combines functionalities from various parts of the system; as such, there are no DMVPN-specific commands. Commands for implementing DMVPN are described in the following guides.

Related commands documented elsewhere	
IPsec	Commands for using the IP Security (IPsec) suite of protocols are described in AT&T Vyatta Network Operating System Services Configuration Guide *and* AT&T Vyatta Network Operating System IPsec Site-to-Site VPN Configuration Guide.
mGRE	Commands for using multipoint Generic Routing Encapsulation (mGRE) are described in AT&T Vyatta Network Operating System Tunnels Configuration Guide.
NHRP	Commands for using the Next Hop Resolution Protocol (NHRP) addressing service are described in AT&T Vyatta Network Operating System Services Configuration Guide.
Routing Protocols	Commands for using the Routing Information Protocol (RIP), RIP next generation, (RIPng), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP) are described in AT&T Vyatta Network Operating System RIP Configuration Guide, AT&T Vyatta Network Operating System RIPng Configuration Guide, AT&T Vyatta Network Operating System OSPF Configuration Guide, *and* AT&T Vyatta Network Operating System BGP Configuration Guide, respectively.

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access