BIG-IP® TMOS®: Tunneling and IPsec

Version 11.6

Table of Contents

Legal Notices

Publication Date

This document was published on July 1, 2015.

Publication Number

MAN-0537-00

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, Application Acceleration Manager, Application Security Manager, APM, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Certified [DESIGN], F5 Networks, F5 SalesXchange [DESIGN], F5 Synthesis, f5 Synthesis, F5 Synthesis [DESIGN], F5 TechXchange [DESIGN], Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, iCall, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, LineRate, LineRate Systems [DESIGN], Local Traffic Manager, LROS, LTM, Message Security Manager, MobileSafe, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, SDAC (except in Japan), SDC, Signalling Delivery Controller, Solutions for an application world, Software Designed Applications Services, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix (except Germany), Traffix [DESIGN] (except Germany), Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WebSafe, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at:
http://www.f5.com/about/guidelines-policies/patents

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes software with glib library utility functions, which is protected under the GNU Public License.

This product includes software with grub2 bootloader functions, which is protected under the GNU Public License.

This product includes software with the Intel Gigabit Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes software with the Intel 10 Gigabit PCI Express Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes software developed by Andrew Tridgell, which is protected under the GNU Public License, copyright ©1992-2000.

This product includes software developed by Jeremy Allison, which is protected under the GNU Public License, copyright ©1998.

This product includes software developed by Guenther Deschner, which is protected under the GNU Public License, copyright ©2008.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes gd-libgd library software developed by the following in accordance with the following copyrights:

- Portions copyright ©1994, 1995, 1996, 1997, 1998, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.
- Portions copyright ©1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.
- Portions relating to GD2 format copyright ©1999, 2000, 2001, 2002 Philip Warner.
- Portions relating to PNG copyright ©1999, 2000, 2001, 2002 Greg Roelofs.
- Portions relating to gdttf.c copyright ©1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).
- Portions relating to gdft.c copyright ©2001, 2002 John Ellson (ellson@lucent.com).
- Portions copyright ©2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 2008 Pierre-Alain Joye (pierre@libgd.org).
- Portions relating to JPEG and to color quantization copyright ©2000, 2001, 2002, Doug Becker and copyright ©1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group.
- Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande. Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This product includes software developed by Oracle America, Inc. Copyright ©2012.

1. Java Technology Restrictions. Licensee shall not create, modify, change the behavior of, or authorize licensees of licensee to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax”, "sun" or similar convention as specified by Oracle in any naming convention designation. In the event that Licensee creates an additional API(s) which: (a) extends the functionality of a Java Environment; and (b) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, Licensee must promptly publish broadly an accurate specification for such API for free use by all developer.
2. Trademarks and Logos. This License does not authorize an end user licensee to use any Oracle America, Inc. name, trademark, service mark, logo or icon. The end user licensee acknowledges that Oracle owns the Java trademark and all Java-related trademarks, logos and icon including the Coffee Cup and Duke ("Java Marks") and agrees to: (a) comply with the Java Trademark Guidelines at http://www.oraclc.com/html/3party.html; (b) not do anything harmful to or inconsistent with Oracle's rights in the Java Marks; and (c) assist Oracle in protecting those rights, including assigning to Oracle any rights acquired by Licensee in any Java Mark.
3. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of your license. Source code may not be redistributed unless expressly provided for in the terms of your license.
4. Third Party Code. Additional copyright notices and license terms applicable to portion of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file.
5. Commercial Features. Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified in Table I-I (Commercial Features In Java SE Product Editions) of tile Software documentation accessible at http://www.oracle.com/technetwork/java/javase/documentation/index.html.

This product includes utilities developed by Linus Torvalds for inspecting devices connected to a USB bus.

This product includes perl-PHP-Serialization software, developed by Jesse Brown, copyright ©2003, and distributed under the Perl Development Artistic License (http://dev.perl.org/licenses/artistic.html).

This product includes software developed by members of the CentOS Project under the GNU Public License, copyright ©2004-2011 by the CentOS Project.

4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes the ixgbevf Intel Gigabit Linux driver, Copyright © 1999 - 2012 Intel Corporation, and distributed under the GPLv2 license, as published by the Free Software Foundation.

This product includes libwebp software. Copyright © 2010, Google Inc. All rights reserved.

This product includes Angular software developed by Google, Inc., http://angulargs.org, copyright © 2010-2012 Google, Inc., and distributed under the MIT license.

This product includes node.js software, copyright © Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes bootstrap software, copyright © 2011-2014 Twitter, Inc., and distributed under the MIT license (http://getbootstrap.com/getting-started/#license-faqs).

This product includes Intel PCM software, copyright © 2009-2013, Intel Corporation All rights reserved. This software is distributed under the OSI BSD license.

This product includes jxrlib software, copyright ©2009 Microsoft Corp. All rights reserved. Distributed under the new BSD license.

This product includes Net-SNMP software, to which one or more of the following copyrights apply:

- Copyright © 1989, 1991, 1992 by Carnegie Mellon University; Derivative Work - 1996, 1998-2000, Copyright © 1996, 1998-2000, The Regents of the University of California. All rights reserved. Distributed under CMU/UCD license (BSD like).
- Copyright © 2001-2003, Networks Associates Technology, Inc. All rights reserved. Distributed under the BSD license.
- Portions of this code are copyright © 2001-2003, Cambridge Broadband Ltd. All rights reserved. Distributed under the BSD license.
- Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved. Distributed under the BSD license.
- Copyright © 2003-2009, Sparta, Inc. All rights reserved. Distributed under the BSD license.
- Copyright © 2004, Cisco, Inc and Information Network Center of Beijing University of Posts and Telecommunications. All rights reserved. Distributed under the BSD license.
- Copyright © 2003Fabasoft R&D Software GmbH & Co KG, oss@fabasoft.com. Distributed under the BSD license.
- Copyright © 2007 Apple Inc. All rights reserved. Distributed under the BSD license.
- Copyright © 2009 ScienceLogic, Inc. All rights reserved. Distributed under the BSD license.

This product includes Racoon 2 software, copyright © 2003-2005 WIDE Project. All rights reserved. Distributed under a BSD-like license.

This product includes node-uuid software, copyright © 2010-2012, Robert Kieffer, and distributed under the MIT license.

This product includes opencsv software, which is distributed under the Apache 2.0 license.

This product includes owasp-jave-encoder software, copyright © 2014, Jeff Ichnowski, and distributed under the New BSD license.

This product may include Intel SDD software subject to the following license; check your hardware specification for details.

1. LICENSE. This Software is licensed for use only in conjunction with Intel solid state drive (SSD) products. Use of the Software in conjunction with non-Intel SSD products is not licensed hereunder. Subject to the terms of this Agreement, Intel grants to You a nonexclusive, nontransferable, worldwide, fully paid-up license under Intel's copyrights to:

   - copy the Software onto a single computer or multiple computers for Your personal, noncommercial use; and
   - make appropriate back-up copies of the Software, for use in accordance with Section 1a) above.

   The Software may contain the software or other property of third party suppliers, some of which may be identified in, and licensed in accordance with, any enclosed "license.txt" file or other text or file.

   Except as expressly stated in this Agreement, no license or right is granted to You directly or by implication, inducement, estoppel or otherwise. Intel will have the right to inspect or have an independent auditor inspect Your relevant records to verify Your compliance with the terms and conditions of this Agreement.

2. RESTRICTIONS. You will not:

   a. copy, modify, rent, sell, distribute or transfer any part of the Software, and You agree to prevent unauthorized copying of the Software; and,
   b. reverse engineer, decompile, or disassemble the Software; and,
   c. sublicense or permit simultaneous use of the Software by more than one user; and,
   d. otherwise assign, sublicense, lease, or in any other way transfer or disclose Software to any third party, except as set forth herein; and,
   e. subject the Software, in whole or in part, to any license obligations of Open Source Software including without limitation combining or distributing the Software with Open Source Software in a manner that subjects the Software or any portion of the Software provided by Intel hereunder to any license obligations of such Open Source Software. "Open Source Software" means any software that requires as a condition of use, modification and/or distribution of such software that such software or other software incorporated into, derived from or distributed with such software:

      a. be disclosed or distributed in source code form; or
      b. be licensed by the user to third parties for the purpose of making and/or distributing derivative works; or
      c. be redistributable at no charge.

   Open Source Software includes, without limitation, software licensed or distributed under any of the following licenses or distribution models, or licenses or distribution models substantially similar to any of the following:

   a. GNU’s General Public License (GPL) or Lesser/Library GPL (LGPL),
   b. the Artistic License (e.g., PERL),
   c. the Mozilla Public License,
   d. the Netscape Public License,
   e. the Sun Community Source License (SCSL),
   f. vi) the Sun Industry Source License (SISL),
   g. vii) the Apache Software license, and
   h. viii) the Common Public License (CPL).

3. OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to materials referenced therein, at any time and without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right or license under Intel patents, copyrights, trademarks, or other intellectual property rights.
4. Entire Agreement. This Agreement contains the complete and exclusive statement of the agreement between You and Intel and supersedes all proposals, oral or written, and all other communications relating to the subject matter of this Agreement. Only a written instrument duly executed by authorized representatives of Intel and You may modify this Agreement.
5. LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety (90) days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.
6. EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for any errors, the accuracy or completeness of any information, text, graphics, links or other materials contained within the Software.
7. LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.
8. TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if You violate its terms. Upon termination, You will immediately destroy the Software or return all copies of the Software to Intel.
9. APPLICABLE LAWS. Claims arising under this Agreement will be governed by the laws of Delaware, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.
10. GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or their successors. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054.

Chapter 1

Configuring Network Virtualization Segments

Overview: Configuring network virtualization tunnels

Large data centers and cloud service providers are benefiting from large scale network virtualization. Network Virtualization provides connectivity in cloud environments by overlaying Layer 2 segments over a Layer 3 infrastructure. The overlay network can be dynamically extended with multiple virtualized networks without affecting the Layer 3 infrastructure. This number of virtualized networks is typically much larger than the number of VLANS the infrastructure can support.

You can configure a BIG-IP® system to function as a gateway in a virtualized network, bridging the data center virtualized networks with the physical network (L2 gateway), or performing routing and higher L4-L7 functionality among virtual networks of different types (L3 gateway). Connecting these networks allows for expansion, and provides a mechanism to streamline the transition of data centers into a virtualized model, while maintaining connectivity.

This illustration shows the BIG-IP system as a network virtualization gateway.

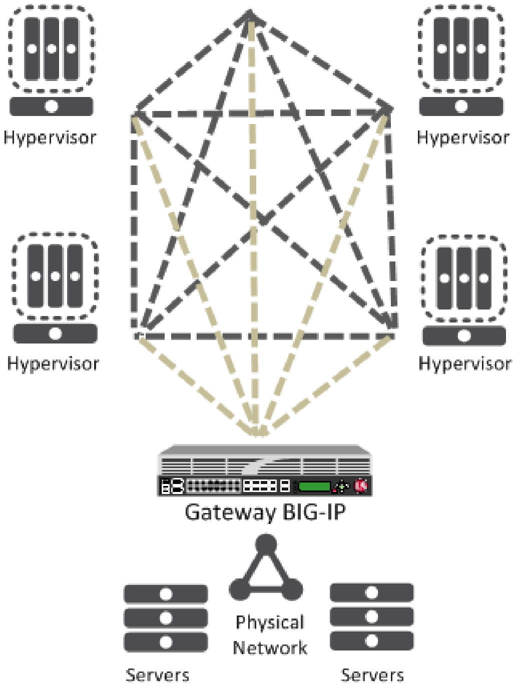


Figure 1: The BIG-IP system as a network virtualization gateway

In a virtualized network, the BIG-IP system needs to learn about other virtualization tunnel endpoints. Each hypervisor has a tunnel endpoint. The hypervisor needs to locate the virtual machines it manages, by maintaining a form of the L2 location records, typically, IP addresses and MAC addresses, virtual network identifiers, and virtual tunnel endpoints.

About network virtualization tunnels on the BIG-IP system

When you configure a BIG-IP® system as a network virtualization gateway, the system represents the connection as a tunnel, which provides a Layer 2 interface on the virtual network. You can use the tunnel interface in both Layer 2 and Layer 3 configurations. After you create the network virtualization tunnels, you can use the tunnels like you use VLANs on a BIG-IP system, such as for routing, assigning self IP addresses, and associating with virtual servers.

Creating a network virtualization tunnel

Creating a network virtualization tunnel on a BIG-IP® system provides an L2 gateway to connect the physical underlay network with a virtual overlay network.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the tunnel profile you created for network virtualization.

   This selection must be a profile based on either the `gre` or `vxlan` parent profile, depending on your virtualized network environment.
4. In the **Local Address** field, type the self IP address of the VLAN through which the remote hypervisor is reachable.
5. For the **Remote Address** list, retain the default selection, **Any**.
6. In the **Key** field, type the VNI (Virtual Network Identifier) to use for the VXLAN tunnel.
7. Click **Finished**.

This tunnel is now available to use in virtualized network routing configurations, depending on how you configure your network.

Virtualized network terminology

These terms are associated with virtualized networks.

forwarding database (FDB)
The *FDB* is the database that contains mappings between the MAC address of each virtual machine and the IP address of the hypervisor machine on which it resides.

L2 gateway
The Layer 2 gateway performs the bridge functionality between VLAN and virtual segments in a virtualized network.

L3 gateway
The Layer 3 gateway performs routing and higher L4-L7 functionality among virtualized network segments of different types.

overlay network
The *overlay network* is a virtual network of VMs built on top of a stable L2-L3 structure. The view from one VM to another is as if they were on the same switch, but, in fact, they could be far afield.

tunnel endpoint
A *tunnel endpoint* originates or terminates a tunnel. In a virtualized network environment, the tunnel IP addresses are part of the L2 underlay network. The same local IP address can be used for multiple tunnels.

underlay network
The *underlay network* is the L2 or L3 routed physical network, a mesh of tunnels.

virtualized network
A *virtualized network* is when you create a virtual L2 or L3 topology on top of a stable physical L2 or L3 network. Connectivity in the virtual topology is provided by tunneling Ethernet frames in IP over the physical network.

VNI

The Virtual Network Identifier (VNI) is also called the VXLAN segment ID. The system uses the VNI to identify the appropriate tunnel.

VSID

The Virtual Subnet Identifier (VSID) is a 24-bit identifier used in an NVGRE environment that represents a virtual L2 broadcast domain, enabling routes to be configured between virtual subnets.

VTEP

The *VXLAN Tunnel Endpoint (VTEP)* originates or terminates a VXLAN tunnel. The same local IP address can be used for multiple tunnels.

VXLAN

Virtual eXtended LAN (VXLAN) is a network virtualization scheme that overlays Layer 2 over Layer 3. VLXAN uses Layer 3 multicast to support the transmission of multicast and broadcast traffic in the virtual network, while decoupling the virtualized network from the physical infrastructure.

VXLAN gateway

A *VXLAN gateway* bridges traffic between VXLAN and non-VXLAN environments. The BIG-IP® system uses a VXLAN gateway to bridge a traditional VLAN and a VXLAN network, by becoming a network virtualization endpoint.

VXLAN header

In addition to the UDP header, encapsulated packets include a *VXLAN header*, which carries a 24-bit VNI to uniquely identify Layer 2 segments within the overlay.

VXLAN segment

A *VXLAN segment* is a Layer 2 overlay network over which VMs communicate. Only VMs within the same VXLAN segment can communicate with each other.

Centralized vs. decentralized models of network virtualization

Using the BIG-IP® system as a network virtualization gateway, you can set up virtualized network segments using either a centralized or decentralized model.

Centralized model

In a centralized model, a network orchestrator or controller manages the virtualized network segments. The orchestrator has full view of VTEPs, L2, and L3 information in the overlay, and is responsible for pushing this information to hypervisors and gateways. Microsoft Hyper-V and VMware NSX environments use this model.

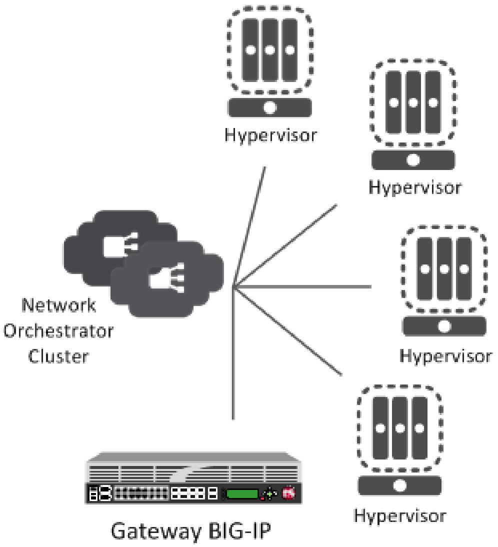


Figure 2: Centralized model of network virtualization

Decentralized model

A decentralized model of network virtualization does not require a network orchestrator or controller. In this model, the router learns the tunnel endpoint and MAC address locations by flooding broadcast, multicast, and unknown destination frames over IP multicast. VMware vSphere 5.1 environments use this model.

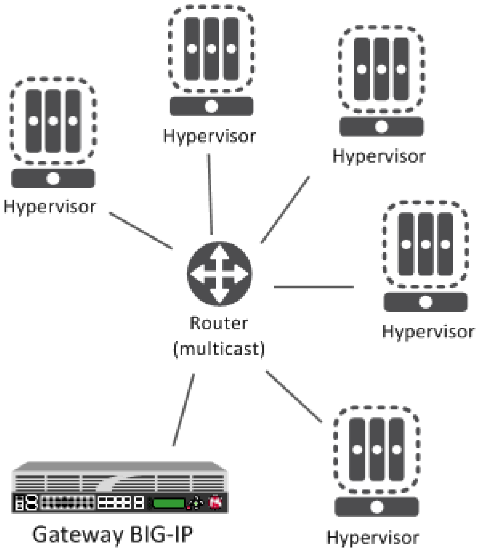


Figure 3: Decentralized model of network virtualization

About network virtualization tunnel types

The BIG-IP® system supports multiple network virtualization tunnel types. You can even combine virtualized network segments based on different tunnel types. This table offers a quick comparison of the tunnel types.

VXLAN (Multicast)	VXLAN (Unicast)	NVGRE	Transparent Ethernet Bridging
Decentralized	Centralized	Centralized	Centralized

VXLAN (Multicast)	VXLAN (Unicast)	NVGRE	Transparent Ethernet Bridging
VMware vSphere 5.1	VMware NSX	Microsoft SCVMM/Hyper-V	OpenStack
VXLAN UDP Encapsulation	VXLAN UDP Encapsulation	GRE-based Encapsulation	GRE-based Encapsulation
24-bit ID	24-bit ID	24-bit ID	32-bit ID
Endpoints discovered dynamically	Endpoints statically configured	Endpoints statically configured	Endpoints statically configured
Floods unknown and broadcast frames using IP multicast.	Can flood using unicast replication.	Does not flood (completely static).	Floods using unicast replication.

About statically configured network virtualization tunnels

For the centralized model, you can use VXLAN (Unicast), NVGRE, or Transparent Ethernet Bridging, depending on the cloud environment. Using an agent or plug-in, or the `tmsh` command-line utility, you can statically configure the FDB and ARP forwarding table entries. Using the `tmsh` command-line utility or browser interface, you can create the network virtualization tunnels, which are managed by the network controller.

Considerations for statically configured network virtualization tunnels

As you configure a BIG-IP® system to be an L2 or L3 gateway for statically configured network virtualization tunnels, keep these considerations in mind.

- The BIG-IP system must be licensed for SDN Services.
- If you have over 2000 connections, set the **Management (MGMT)** setting on the Resource Provisioning screen is to **Large** (**System** > **Resource Provisioning**).

Examples for manually populating L2 location records

Using the `tmsh` command-line utility, you can add static FDB records and ARP entries for each virtual tunnel endpoint.

- Add static FDB (forwarding database) entries to associate MAC addresses with specified tunnel endpoints. For example, the following command creates an FDB entry that associates the MAC address `00:01:02:03:04:05` with the tunnel endpoint `10.1.1.1` of the tunnel `vxlan0`.

```
# tmsh modify net fdb tunnel vxlan0 records add {
    00:01:02:03:04:05 { endpoint 10.1.1.1 } }
```

- Delete a MAC address from an FDB entry.

```
# tmsh modify net fdb tunnel vxlan0 records del { 00:01:02:03:04:05 }
```

- Add an IP address to a MAC address in the ARP table.

```
# tmsh modify net arp 10.3.3.1 { ip-address 10.3.3.1 mac-address 00:01:02:03:04:05 }
}
```

Using the iControl/REST API, you can program a network controller to build and maintain network virtualization tunnels. This example adds an entry to the FDB table that associates the MAC address `00:01:02:03:04:05` with the tunnel endpoint `10.1.1.2` of the tunnel `vxlan0-tunnel`.

```
$ curl -u admin:f5site02 -H "Content-Type:=application/json" -k -X PUT
'https://172.30.69.69/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel' -d
'{"kind":"tm:net:fdb:tunnel:tunnelstate","name":"vxlan0-tunnel","partition":"Common",
"fullPath":"/Common/vxlan0-tunnel","generation":1,
"selfLink":"https://localhost/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel?
ver=11.5.0","records":[{"name":"00:01:02:03:04:05",
"endpoint":"10.1.1.2"}]}'  |python -m json.tool
{
    "fullPath": "/Common/vxlan0-tunnel",
    "generation": 1,
    "kind": "tm:net:fdb:tunnel:tunnelstate",
    "name": "vxlan0-tunnel",
    "partition": "Common",
    "records": [
        {
            "endpoint": "10.1.1.2",
            "name": "00:01:02:03:04:05"
        }
    ],
    "selfLink": "https://localhost/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel?ver=11.5.0"
}
```

Sample NVGRE configuration using tmsh

This listing example illustrates the steps for creating a routing configuration that includes an NVGRE tunnel on the BIG-IP® system. F5 Networks provides an API for you to configure the F5 SCVMM Gateway Provider plug-in to build and manage NVGRE tunnels.

```
create net vlan wan {
    interfaces add { 1.1 }
    mtu 1550
}
create net self 10.1.1.1/24 {
    address 10.1.1.1/24
    vlan wan
}
create net tunnels gre nvgre {
    encapsulation nvgre
}
create net tunnels tunnel nvgre5000 {
    local-address 10.1.1.1
    remote-address any
    profile nvgre
    key 5000
}
create net vlan legacy5000 {
    interfaces add { 2.1 }
}
create net route-domain 5000 {
    id 5000
    vlans add { nvgre5000 legacy5000 }
}
create net self 10.3.3.1%5000/24 {
    address 10.3.3.1%5000/24
    vlan nvgre5000
}
```

```
create net self 10.4.4.1%5000/24 {
    address 10.4.4.1%5000/24
    vlan legacy5000
}
create net route 10.5.5.0%5000/24 {
    network 10.5.5.0%5000/24
    gw 10.3.3.2%5000
}
create net route 10.6.6.0%5000/24 {
    network 10.6.6.0%5000/24
    gw 10.3.3.3%5000
}
modify net fdb tunnel nvgre5000 {
    records add {
        00:FF:0A:03:03:02 { endpoint 10.1.2.1 }
        00:FF:0A:03:03:03 { endpoint 10.1.3.1 }
    }
}
create net arp 10.3.3.2%5000 {
    mac-address 00:FF:0A:03:03:02
}
create net arp 10.3.3.3%5000 {
    mac-address 00:FF:0A:03:03:03
}
```

Sample VXLAN unicast configuration using tmsh

This example listing illustrates the steps for creating a routing configuration that includes a VXLAN tunnel on the BIG-IP® system. This configuration adds the tunnel to a route domain. You can use the iControl/REST API to configure a network controller to build and manage VXLAN (unicast) tunnels.

```
create net vlan wan {
    interfaces add { 1.1 }
    mtu 1550
}
create net self 10.1.1./24 {
    address 10.1.1.1/24
    vlan wan
}
create net tunnels vxlan vxlan-static {
    flooding-type none
}
create net tunnels tunnel vxlan5000 {
    local-address 10.1.1.1
    remote-address any
    profile vxlan-static
    key 5000
}
create net vlan legacy5000 {
    interfaces add { 2.1 }
}
create net route-domain 5000 {
    id 5000
    vlans add {vxlan5000 legacy5000 }
}
create net self 10.3.3.1%5000/24 {
    address 10.3.3.1%5000/24
    vlan vxlan5000
}
create net self 10.4.4.1%5000/24 {
    address 10.4.4.1%5000/24
    vlan legacy5000
}
create net route 10.5.5.0%5000/24 {
    network 10.5.5.0%5000/24
    gw 10.3.3.2%5000
}
create net route 10.6.6.0%5000/24 {
    network 10.6.6.0%5000/24
    gw 10.3.3.3%5000
}
modify net fdb tunnel vxlan5000 {
    records add {
        00:FF:0A:03:03:02 { endpoint 10.1.2.1 }
        00:FF:0A:03:03:03 { endpoint 10.1.3.1 }
    }
}
create net arp 10.3.3.2%5000 {
    mac-address 00:FF:0A:03:03:02
}
create net arp 10.3.3.3%5000 {
    mac-address 00:FF:0A:03:03:03
}
}
```

Sample command for virtual server to listen on a VXLAN tunnel

An alternative for including a network virtualization tunnel in a routing configuration is to create a virtual server that listens for the tunnel traffic, such as in the following example.

```
# tmsh create ltm virtual http_virtual destination 10.3.3.15%5000:http ip-protocol tcp vlans
 add { vxlan5000 }
```

The code in this example creates a virtual server `http_virtual` that listens for traffic destined for the IP address `10.3.3.15`on the tunnel named `vxlan5000`.

Commands for viewing tunnel statistics

You can use the `tmsh` command-line utility to view tunnel statistics, listing either all the tunnels on the BIG-IP® system or statistics about a particular tunnel.

View per-tunnel statistics:

```
# tmsh show net tunnels tunnel
```

View static and dynamic FDB entries:

```
# tmsh show net fdb tunnel
```

About VXLAN multicast configuration

In a VMware vSphere 5.1 environment, you can configure VXLAN without knowing all the remote tunnel endpoints. The BIG-IP® system uses multicast flooding to learn unknown and broadcast frames. VXLAN can extend the virtual network across a set of hypervisors, providing L2 connectivity among the hosted virtual machines (VMs). Each hypervisor represents a VXLAN tunnel endpoint (VTEP). In this environment, you can configure a BIG-IP system as an L2 VXLAN gateway device to terminate the VXLAN tunnel and forward traffic to and from a physical network.

Task summary

About bridging VLAN and VXLAN networks

You can configure Virtual eXtended LAN (VXLAN) on a BIG-IP® system to enable a physical VLAN to communicate with virtual machines (VMs) in a virtual network.

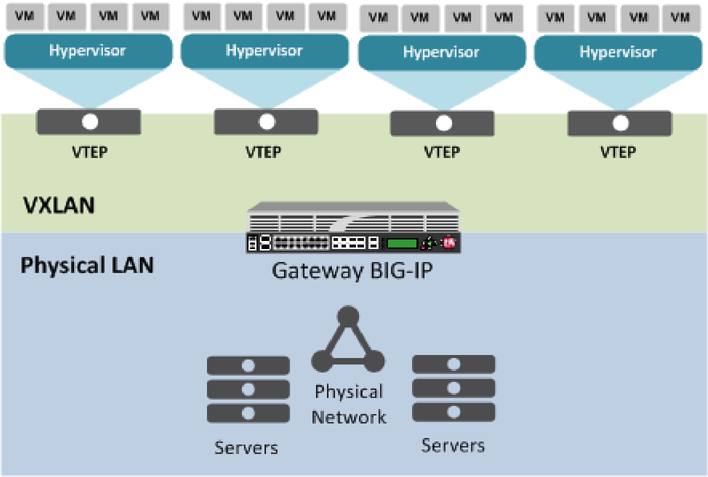


Figure 4: The VXLAN gateway

When you configure a BIG-IP system as an L2 VXLAN gateway, the BIG-IP system joins the configured multicast group, and can forward both unicast and multicast or broadcast frames on the virtual network. The BIG-IP system learns about MAC address and VTEP associations dynamically, thus avoiding unnecessary transmission of multicast traffic.

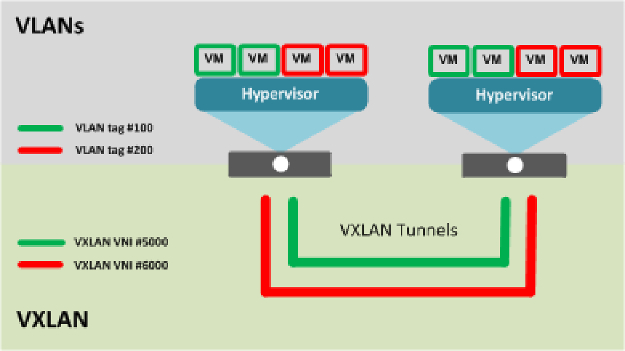


Figure 5: Multiple VXLAN tunnels

Considerations for configuring multicast VXLAN tunnels

As you configure VXLAN on a BIG-IP® system, keep these considerations in mind.

- If you configure the BIG-IP device as a bridge between physical VLANs and a VXLAN tunnel, the number of virtualized network segments in the overlay is limited to the maximum number of physical VLANs (4094). This limitation does not apply to Layer 3 configurations.
- You need to configure a separate tunnel for each VNI. The tunnels can have the same local and remote endpoint addresses.
- For the Layer 2 network, you must ensure a loop-free topology.
- Do not modify the configuration of a VXLAN tunnel after it is created. Instead, delete the existing tunnel and create a new one.

Task summary

Before you configure VXLAN, ensure that these conditions are met:

- The BIG-IP® system must be licensed for SDN Services.
- Network connectivity exists between the BIG-IP system and the hypervisors.
- If you have over 2000 tunnels, the **Management (MGMT)** setting on the Resource Provisioning screen is set to **Large** (**System** > **Resource Provisioning**).

Task list

About VXLAN multicast configuration
Creating a multicast VXLAN tunnel
Creating a bridge between VXLAN and non-VXLAN networks

Creating a multicast VXLAN tunnel

Creating a VXLAN multicast tunnel on a BIG-IP® system provides an L2 VXLAN gateway to connect the physical network with a virtualized network.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **vxlan**.

   This setting tells the system which tunnel profile to use. The system-supplied VXLAN profile specifies port `4789`. To change the port number, you can create a new VXLAN profile, which then appears in this list.
4. In the **Local Address** field, type the self IP address of the VLAN through which the remote hypervisor is reachable.
5. In the **Remote Address** field, type the multicast group address associated with the VXLAN segment.
6. In the **Key** field, type the VNI (Virtual Network Identifier) to use for the VXLAN tunnel.
7. Click **Finished**.

Creating a bridge between VXLAN and non-VXLAN networks

Before you begin this task, verify that a VXLAN multicast tunnel exists on the BIG-IP® system.

You can create a VLAN group to bridge the traffic between a VXLAN overlay network (Layer 3) and a non-VXLAN (Layer 2) network.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In then **Name** field, type a unique name for the VLAN group.
4. For the **VLANs** setting, select the VLAN that connects to the non-VXLAN Layer-2 network and the VXLAN tunnel you created, and using the Move button, move your selections from the **Available** list to the **Members** list.
5. Click **Finished**.

About configuring VXLAN tunnels on high availability BIG-IP device pairs

By default, the BIG-IP® system synchronizes all existing tunnel objects in its config sync operation. This operation requires that the local IP address of a tunnel be set to a floating self IP address. In a high availabilty (HA) configuration, any tunnel with a floating local IP address would be available only on the active device, which would prevent some features, such as health monitors, from using the tunnel on the standby device. To make a tunnel available on both the active and standby devices, you need to set the local IP address to a non-floating self IP address, which then requires that you exclude tunnels from the config sync operation. To disable the synchronization of tunnel objects, you can set a `bigdb` variable on both devices.

Disabling config sync for tunnels

In certain cases, you might want to disable config sync behavior for tunnels, such as when you need to make VXLAN tunnels functional on all devices in a BIG-IP® device group configured for high availability. The tunnel config sync setting applies to all tunnels created on the BIG-IP device.

Important: *Disable config sync on both the active and standby devices before you create any tunnels.*

1. Log in to the`tmsh` command-line utility for the BIG-IP system.
2. Determine whether the variable is already disabled, by typing this command.

   ```
   tmsh list sys db iptunnel.configsync value
   ```
3. Disable the variable.

   ```
   tmsh modify sys db iptunnel.configsync value disable
   ```
4. Save the configuration.

   ```
   tmsh save sys config
   ```
5. F5 recommends that you reboot both the active and standby devices.

Now you can create tunnels with non-floating local IP addresses on both the active and standby devices.

Chapter 2

Creating IP Tunnels

About IP tunnels

Using F5® tunneling technologies, you can set up tunneling from devices on different Layer 2 networks, or scale multi-site data centers over Layer 3 pathways. When you know the IP address of the devices at both ends of the tunnel, you can create a point-to-point encapsulation tunnel between a BIG-IP® system and another device. When multiple devices feed into a BIG-IP system, you can create a tunnel by specifying only the IP address on the BIG-IP device.

The BIG-IP system provides the following tunneling types, available using the browser-based Configuration utility or the Traffic Management shell (`tmsh`) command-line utility, and iControl®.

- EtherIP
- FEC
- GRE
- IPIP
    - DS-Lite
    - IPv4IPv4
    - IPv4IPv6
    - IPv6IPv4
    - IPv6IPv6
- PPP
- VXLAN
- WCCPGRE

For information about deploying some of these tunneling types, consult additional F5 Networks documentation, including CGNAT (DS-Lite), acceleration (FEC), and TMOS (VXLAN). Licensing restrictions apply.

About point-to-point tunnels

Point-to-point IP encapsulation tunnels carry traffic through a routed network between known devices. For example, you can create a GRE tunnel to connect a BIG-IP® system to a remotely located pool member.

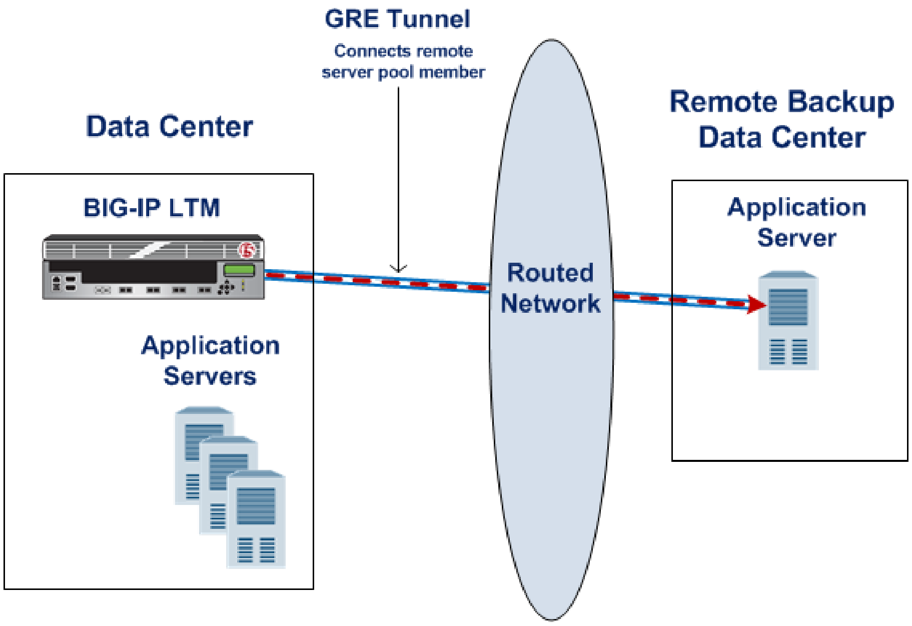


Figure 6: Illustration of a point-to-point GRE tunnel

Task summary

Creating a point-to-point IP tunnel
Assigning a self IP address to an IP tunnel endpoint
Routing traffic through an IP tunnel interface

Creating a point-to-point IP tunnel

To create a point-to-point tunnel, you specify the encapsulation protocol and the IP addresses of the devices at both ends of the tunnel.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the type that corresponds to the encapsulation protocol you want to use.

   The selection **ipip** is the same as **ip4ip4**, but **ipip** is compatible with configurations from an earlier release.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, select **Specify**, and type the IP address of the device at the other end of the tunnel.
6. Click **Finished**.

After you complete this task, traffic is encapsulated using the protocol you specified between the BIG-IP system and the remote device you specified.

The BIG-IP®system requires that tunnels used as routes have a self IP address associated with the tunnel itself, distinct from the self IP address configured as a tunnel endpoint. After configuring a self IP address, you can configure routes that use the tunnel as a resource.

Assigning a self IP address to an IP tunnel endpoint

Ensure that you have created an IP tunnel before starting this task.

Self IP addresses can enable the BIG-IP® system, and other devices on the network, to route application traffic through the associated tunnel, similar to routing through VLANs and VLAN groups.

***Note:** If the other side of the tunnel needs to be reachable, make sure the self IP addresses that you assign to both sides of the tunnel are in the same subnet.*

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type the IP address of the tunnel.

   The system accepts IPv4 and IPv6 addresses.

   ***Note:** This is not the same as the IP address of the tunnel local endpoint.*
5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select the tunnel with which to associate this self IP address.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

Assigning a self IP to a tunnel ensures that the tunnel appears as a resource for routing traffic.

To direct traffic through the tunnel, add a route for which you specify the tunnel as the resource.

Routing traffic through an IP tunnel interface

Before starting this task, ensure that you have created an IP tunnel, and have assigned a self IP address to the tunnel.

You can route traffic through a tunnel interface, much like you use a VLAN or VLAN group.

1. On the Main tab, click **Network** > **Routes**.
2. Click **Add**.
   The New Route screen opens.
3. In the **Name** field, type a unique user name.

   This name can be any combination of alphanumeric characters, including an IP address.
4. In the **Destination** field, type the destination IP address for the route.
5. In the **Netmask** field, type the network mask for the destination IP address.
6. From the **Resource** list, select **Use VLAN/Tunnel**.
7. From the **VLAN/Tunnel** list, select a tunnel name.
8. Click **Finished**.

The system now routes traffic destined for the IP address you specified through the tunnel you selected.

Example of a point-to-point IP tunnel configuration

This illustration is an example of a point-to-point IP tunnel configuration showing IP addresses. Note that the tunnel local endpoint address is different from the tunnel self IP address.

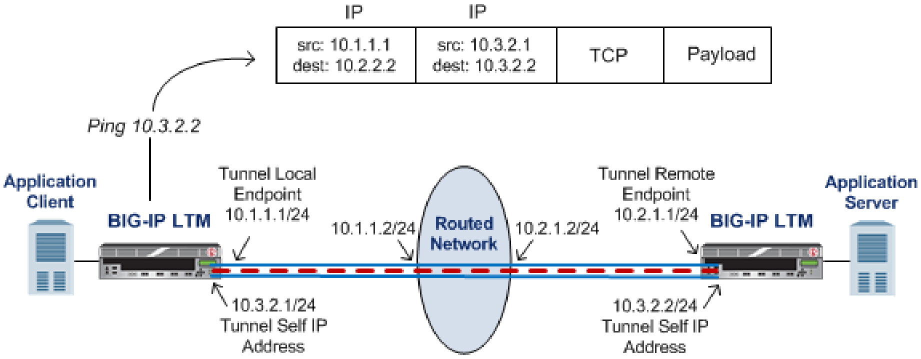


Figure 7: Illustration of a point-to-point IP tunnel configuration

About tunnels between the BIG-IP system and other devices

In a network that has multiple devices connected to a BIG-IP® system, you can create an IPIP or GRE encapsulation tunnel between the BIG-IP system and the remote devices without having to specify a remote (or source) IP address for every device. The use cases include situations where the source IP address is unknown or difficult to discover.

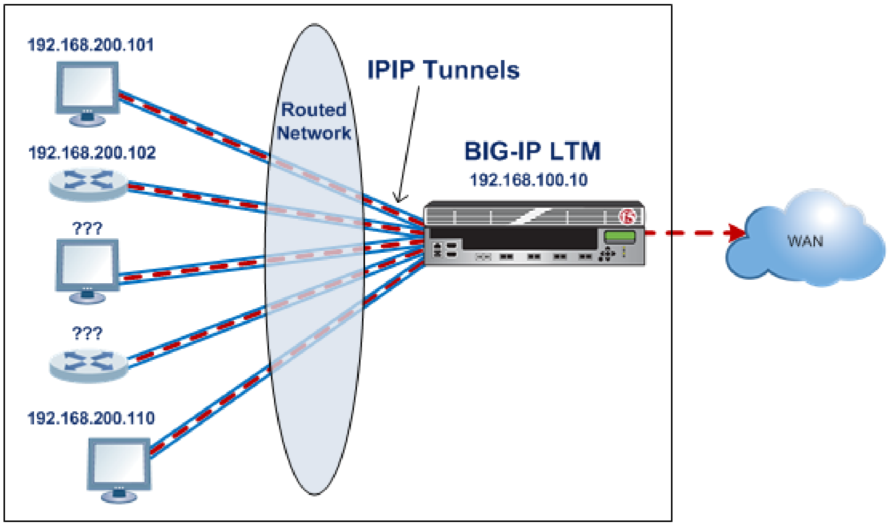


Figure 8: Illustration of an IPIP tunnel between a BIG-IP system and multiple unspecified devices

Creating an encapsulation tunnel between a BIG-IP device and multiple devices

You can create a tunnel between a BIG-IP® system and multiple remote devices without having to specify a remote (or source) IP address for every device.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the type that corresponds to the encapsulation protocol you want to use.

   The selection **ipip** is the same as **ip4ip4**, but **ipip** is compatible with configurations from an earlier release.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, retain the default selection, **Any**.

   This entry means that you do not have to specify the IP address of the remote end of the tunnel, which allows multiple devices to use the same tunnel.
6. Click **Finished**.

When the BIG-IP system receives an encapsulated packet, the system decapsulates the packet, regardless of the source address, and re-injects it into the IP stack, thus allowing the inner IP address to be associated with a virtual server.

If you are configuring routes that use the tunnel as a resource, you must also assign a self IP address to the tunnel itself, which is different from the tunnel local endpoint IP address.

About transparent tunnels

You can create transparent tunnels when you want to inspect and/or manipulate encapsulated traffic that is flowing through a BIG-IP® system. The BIG-IP system terminates the tunnel, while presenting the illusion that the traffic flows through the device unchanged. In this case, the BIG-IP device appears as if it were an intermediate router that simply routes IP traffic through the device.

The transparent tunnel feature enables redirection of traffic based on policies. For example, service providers can redirect traffic with transparent tunnels to apply classification and bandwidth management policies using Policy Enforcement Manager™. To handle payload inspection and manipulation, you can create a policy in the form of a virtual server that accepts encapsulated packets. In the absence of a policy, the tunnel simply traverses the BIG-IP device.

Transparent tunnels are available for IPIP and GRE encapsulation types, with only one level of encapsulation.

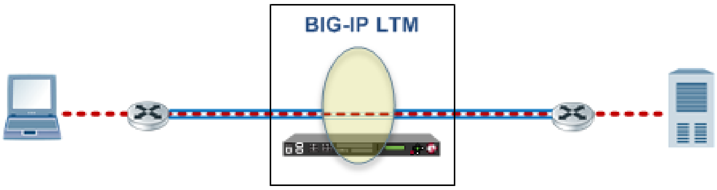


Figure 9: Illustration of a transparent tunnel

When the BIG-IP system receives an encapsulated packet from a transparent tunnel, the system decapsulates the packet, and re-injects it into the IP stack, where a virtual server can pick up the packet to apply a policy or rule. After applying the policy or rule, the BIG-IP can re-encapsulate the packet and route it, as if the packet had transited the BIG-IP unperturbed.

Creating a transparent tunnel

You can create transparent tunnels to inspect and modify tunneled traffic flowing through a BIG-IP® system.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **ipip** or **gre**.

   The **ipip** selection can also be one of the IPIP variations: **ip4ip4**, **ip4ip6**, **ip6ip4**, or **ip6ip6**.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, retain the default selection, **Any**.

   This entry means that you do not have to specify the IP address of the remote end of the tunnel, which allows multiple devices to use the same tunnel.
6. Select the **Transparent** check box.
7. Click **Finished**.

Traffic flowing through the transparent tunnel you created is available for inspection and modification, before continuing to its destination.

After you create a transparent tunnel, additional configuration is required to process the traffic, such as creating a virtual server to intercept the traffic, and using Policy Enforcement Manager™ to apply classification and bandwidth management policies.

Chapter 3

Configuring an EtherIP Tunnel

Overview: Preserving BIG-IP connections during live virtual machine migration

In some network configurations, the BIG-IP® system is configured to send application traffic to destination servers that are implemented as VMware® virtual machines (VMs). These VMs can undergo live migration, using VMware vMotion, across a wide area network (WAN) to a host in another data center. Optionally, an iSession® tunnel could provide WAN optimization.

To preserve any existing connections between the BIG-IP system and a virtual machine while the virtual machine migrates to another data center, you can create an EtherIP tunnel.

An *EtherIP tunnel* is an object that you create on each of two BIG-IP systems that sit on either side of a WAN. The EtherIP tunnel uses the industry-standard EtherIP protocol to tunnel Ethernet and IEEE 802.3 media access control (MAC) frames across an IP network. The two EtherIP tunnel objects together form a tunnel that logically connects two data centers. When the application traffic that flows between one of the BIG-IP systems and the VM is routed through the EtherIP tunnel, connections are preserved during and after the VM migration.

After you have configured the BIG-IP system to preserve connections to migrating VMs, you can create a Virtual Location monitor for the pool. A *Virtual Location* monitor ensures that the BIG-IP system sends connections to a local pool member rather than a remote pool one, when some of the pool members have migrated to a remote data center.

Tip: *The BIG-IP system that is located on each end of an EtherIP tunnel can be part of a redundant system configuration. Make sure that both units of any redundant system configuration reside on the same side of the tunnel.*

Illustration of EtherIP tunneling in a VMotion environment

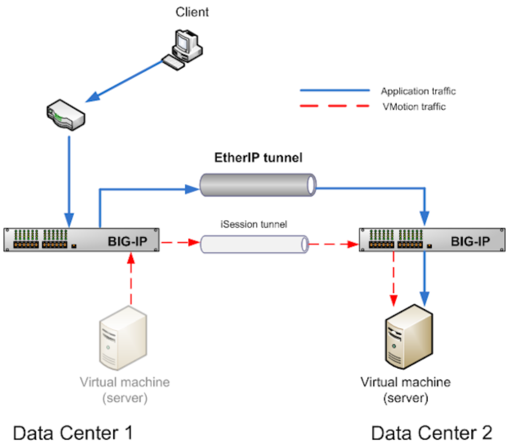


Figure 10: EtherIP tunneling in a VMware VMotion environment

Task summary

Implement an EtherIP tunneling configuration to prevent the BIG-IP® system from dropping existing connections to migrating virtual machines in a VMware vMotion environment.

Important: *Perform these tasks on the BIG-IP system in both the local data center and the remote data center.*

Task List

Creating a VLAN

Creating an EtherIP tunnel object

Creating a VLAN group

Creating a self IP address

Creating a self IP for a VLAN group

Creating a Virtual Location monitor

Syncing the BIG-IP configuration to the device group

Creating a VLAN

VLANs represent a logical collection of hosts that can share network resources, regardless of their physical location on the network. You create a VLAN to associate physical interfaces with that VLAN.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. From the **Customer Tag** list:
   a) Retain the default value of **None** or select **Specify**.
   b) If you chose **Specify** in the previous step, type a numeric tag, from 1-4094, for the VLAN.

   The customer tag specifies the inner tag of any frame passing through the VLAN.
6. For the **Interfaces** setting:
   a) From the **Interface** list, select an interface number.
   b) From the **Tagging** list, select **Tagged** or **Untagged**.

      Select **Tagged** when you want traffic for that interface to be tagged with a VLAN ID.
   c) If you specified a numeric value for the **Customer Tag** setting and from the **Tagging** list you selected **Tagged**, then from the **Tag Mode** list, select a value.
   d) Click **Add**.
   e) Repeat these steps for each interface that you want to assign to the VLAN.
7. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.

8. In the **MTU** field, retain the default number of bytes (**1500**).
9. From the **Configuration** list, select **Advanced**.
10. If you want to base redundant-system failover on VLAN-related events, select the **Fail-safe** check box.
11. From the **Auto Last Hop** list, select a value.
12. From the **CMP Hash** list, select a value.
13. To enable the **DAG Round Robin** setting, select the check box.
14. Configure the sFlow settings or retain the default values.
15. Click **Finished**.
    The screen refreshes, and displays the new VLAN in the list.

Creating an EtherIP tunnel object

Before you perform this task, you must know the self IP address of the instance of the VLAN that exists, or will exist, on the BIG-IP® system in the other data center.

The purpose of an EtherIP tunnel that contains an EtherIP type of profile is to enable the BIG-IP system to preserve any current connections to a server that is using VMware vMotion for migration to another data center.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Profile** list, select **etherip**.
4. In the **Local Address** field, type the self IP address of the local BIG-IP system.
5. In the **Remote Address** field, type the self IP address of the remote BIG-IP system.
6. Click **Finished**.

Creating a VLAN group

VLAN groups consolidate Layer 2 traffic from two or more separate VLANs.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In then **Name** field, type a unique name for the VLAN group.
4. For the **VLANs** setting, select the EtherIP tunnel that you created (which appears in the VLAN list) and the VLAN that connects to the host where the VMs exist, and using the Move button (<<), move your selections from the **Available** list to the **Members** list.
5. From the **Transparency Mode** list, select **Transparent**.
6. Select the **Bridge All Traffic** check box if you want the VLAN group to forward all frames, including non-IP traffic.

   The default setting is disabled (not selected).
7. Retain the **Bridge in Standby** check box selection if you want the VLAN group to forward frames, even when the system is the standby unit of a redundant system.
8. Click **Finished**.

Creating a self IP address

Before you create a self IP address, ensure that you have created a VLAN that you can associate with the self IP address.

A self IP address enables the BIG-IP® system and other devices on the network to route application traffic through the associated VLAN or VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type an IPv4 or IPv6 address.

   This IP address should represent the address space of the VLAN that you specify with the **VLAN/Tunnel** setting.
5. In the **Netmask** field, type the full network mask for the specified IP address.
6. From the **VLAN/Tunnel** list, select the VLAN to associate with this self IP address.
   - On the internal network, select the internal or high availability VLAN that is associated with an internal interface or trunk.
   - On the external network, select the external VLAN that is associated with an external interface or trunk.
7. From the **Port Lockdown** list, select **Allow Default**.
8. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

After you perform this task, the BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Creating a self IP for a VLAN group

Before you create a self IP address, ensure that you have created at least one VLAN group.

You perform this task to create a self IP address for a VLAN group. The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP® system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this procedure, the path to the IP network `10.0.0.0` is actually through more than one VLAN. As IP routers are designed to have only one physical route to a network, a routing conflict can occur. With a self IP address on the BIG-IP system, you can resolve the routing conflict by associating a self IP address with the VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type an IPv4 address.

   This IP address should represent the address space of the VLAN group that you specify with the **VLAN/Tunnel** setting.
5. In the **Netmask** field, type the network mask for the specified IP address.
   For this example, type `255.255.255.0`.

6. From the **VLAN/Tunnel** list, select the VLAN group with which to associate this self IP address.
7. From the **Port Lockdown** list, select **Allow Default**.
8. Click **Finished**.

Creating a Virtual Location monitor

When the BIG-IP® system is directing application traffic to pool members that are implemented as virtual machines, you should configure a Virtual Location type of monitor on the BIG-IP system. A *Virtual Location* monitor determines if a pool member is local to the data center or remote, and assigns a priority group to the pool member accordingly. The monitor assigns remote pool members a lower priority than local members, thus ensuring that the BIG-IP directs application requests to local pool members whenever possible.

1. On the Main tab, click **Local Traffic** > **Monitors**.
   The Monitor List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. Type `my_virtual_location_monitor` in the **Name** field.
4. From the **Type** list, select **Virtual Location**.
5. From the **Configuration** list, select **Advanced**.
6. Retain the default value (in seconds) of `5` in the **Interval** field.
7. Retain the default value of `Disabled` in the **Up Interval** list.
8. Retain the default value (in seconds) of `0` in the **Time Until Up** field.
9. Retain the default value (in seconds) of `16` in the **Timeout** field.
10. Type the name of the pool that you created prior to configuring EtherIP tunneling in the **Pool Name** field.
11. Click **Finished**.

After configuring the Virtual Location monitor, the BIG-IP system assigns each member of the designated pool a priority group value to ensure that incoming connections are directed to a local pool member whenever possible.

F5 Networks recommends that you verify that BIG-IP® Global Traffic Manager™ (GTM™) has automatically assigned a BIG-IP type of monitor to BIG-IP® Local Traffic Manager™ (LTM®). A BIG-IP type of monitor can use the priority group assigned to each pool member to retrieve a `gtm_score` value.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.

The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.

3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Implementation result

After you configure EtherIP tunneling on the BIG-IP system, you must perform the same configuration procedure on the BIG-IP system in the remote data center to fully establish the EtherIP tunnel.

After the tunnel is established, the BIG-IP system preserves any open connections to migrating (or migrated) virtual machine servers.

Chapter 4

Securing EtherIP Tunnel Traffic with IPsec

Overview: Securing EtherIP tunnel traffic with IPsec

You can use the IPsec protocol to secure EtherIP tunnel traffic that is undergoing live migration across a wide area network (WAN) using VMware vMotion. The EtherIP tunnel preserves any existing connections between the BIG-IP® system and a virtual machine while the virtual machine migrates to another data center. Adding IPsec to this configuration involves adding an IPsec traffic selector on each side of the IPsec tunnel. Those traffic selectors have the same source and destination IP addresses as the EtherIP tunnel.

Important: *Perform these tasks on the BIG-IP system in both the local data center and the remote data center.*

Task List

Creating a VLAN
Creating an EtherIP tunnel object
Creating a VLAN group
Creating a self IP address
Creating a self IP for a VLAN group
Creating a custom IPsec policy for EtherIP tunnel traffic
Creating an IPsec traffic selector for EtherIP traffic

Creating a VLAN

VLANs represent a logical collection of hosts that can share network resources, regardless of their physical location on the network. You create a VLAN to associate physical interfaces with that VLAN.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. From the **Customer Tag** list:
   a) Retain the default value of **None** or select **Specify**.
   b) If you chose **Specify** in the previous step, type a numeric tag, from 1-4094, for the VLAN.

   The customer tag specifies the inner tag of any frame passing through the VLAN.
6. For the **Interfaces** setting:
   a) From the **Interface** list, select an interface number.
   b) From the **Tagging** list, select **Tagged** or **Untagged**.

      Select **Tagged** when you want traffic for that interface to be tagged with a VLAN ID.
   c) If you specified a numeric value for the **Customer Tag** setting and from the **Tagging** list you selected **Tagged**, then from the **Tag Mode** list, select a value.
   d) Click **Add**.

e) Repeat these steps for each interface that you want to assign to the VLAN.

7. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.
8. In the **MTU** field, retain the default number of bytes (**1500**).
9. From the **Configuration** list, select **Advanced**.
10. If you want to base redundant-system failover on VLAN-related events, select the **Fail-safe** check box.
11. From the **Auto Last Hop** list, select a value.
12. From the **CMP Hash** list, select a value.
13. To enable the **DAG Round Robin** setting, select the check box.
14. Configure the sFlow settings or retain the default values.
15. Click **Finished**.
    The screen refreshes, and displays the new VLAN in the list.

Creating an EtherIP tunnel object

Before you perform this task, you must know the self IP address of the instance of the VLAN that exists, or will exist, on the BIG-IP® system in the other data center.

The purpose of an EtherIP tunnel that contains an EtherIP type of profile is to enable the BIG-IP system to preserve any current connections to a server that is using VMware vMotion for migration to another data center.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Profile** list, select **etherip**.
4. In the **Local Address** field, type the self IP address of the local BIG-IP system.
5. In the **Remote Address** field, type the self IP address of the remote BIG-IP system.
6. Click **Finished**.

Creating a VLAN group

VLAN groups consolidate Layer 2 traffic from two or more separate VLANs.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In then **Name** field, type a unique name for the VLAN group.
4. For the **VLANs** setting, select the EtherIP tunnel that you created (which appears in the VLAN list) and the VLAN that connects to the host where the VMs exist, and using the Move button (<<), move your selections from the **Available** list to the **Members** list.
5. From the **Transparency Mode** list, select **Transparent**.
6. Select the **Bridge All Traffic** check box if you want the VLAN group to forward all frames, including non-IP traffic.

   The default setting is disabled (not selected).

7. Retain the **Bridge in Standby** check box selection if you want the VLAN group to forward frames, even when the system is the standby unit of a redundant system.
8. Click **Finished**.

Creating a self IP address

Before you create a self IP address, ensure that you have created a VLAN that you can associate with the self IP address.

A self IP address enables the BIG-IP® system and other devices on the network to route application traffic through the associated VLAN or VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type an IPv4 or IPv6 address.

   This IP address should represent the address space of the VLAN that you specify with the **VLAN/Tunnel** setting.
5. In the **Netmask** field, type the full network mask for the specified IP address.
6. From the **VLAN/Tunnel** list, select the VLAN to associate with this self IP address.
   - On the internal network, select the internal or high availability VLAN that is associated with an internal interface or trunk.
   - On the external network, select the external VLAN that is associated with an external interface or trunk.
7. From the **Port Lockdown** list, select **Allow Default**.
8. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

After you perform this task, the BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Creating a self IP for a VLAN group

Before you create a self IP address, ensure that you have created at least one VLAN group.

You perform this task to create a self IP address for a VLAN group. The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP® system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this procedure, the path to the IP network `10.0.0.0` is actually through more than one VLAN. As IP routers are designed to have only one physical route to a network, a routing conflict can occur. With a self IP address on the BIG-IP system, you can resolve the routing conflict by associating a self IP address with the VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type an IPv4 address.

This IP address should represent the address space of the VLAN group that you specify with the **VLAN/Tunnel** setting.

5. In the **Netmask** field, type the network mask for the specified IP address.
   For this example, type `255.255.255.0`.
6. From the **VLAN/Tunnel** list, select the VLAN group with which to associate this self IP address.
7. From the **Port Lockdown** list, select **Allow Default**.
8. Click **Finished**.

Creating a custom IPsec policy for EtherIP tunnel traffic

When you use IPsec to secure EtherIP tunnel traffic, you must create a custom IPsec policy for the traffic selector to use.

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
5. In the **Tunnel Local Address** field, type an IP address.

   This IP address must match the local address of the EtherIP tunnel and the source IP address of the associated traffic selector.
6. In the **Tunnel Remote Address** field, type an IP address.

   This IP address must match the remote address of the EtherIP tunnel and the destination IP address of the associated traffic selector.
7. Click **Finished**.
   The screen refreshes and displays the new IPsec policy in the list.

Creating an IPsec traffic selector for EtherIP traffic

Before you start this task, make sure that you have created a custom IPsec policy to use with this traffic selector.

When you use IPsec to secure EtherIP tunnel traffic, you must create an IPsec traffic selector at each end of the IPsec tunnel to capture the EtherIP traffic.

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. For the **Source IP Address or CIDR** setting, type an IP address.

   This IP address must match the IP address specified for the **Tunnel Local Address** in the selected IPsec policy.
5. For the **Destination IP Address or CIDR** setting, type an IP address.

   This IP address must match the IP address specified for the **Tunnel Remote Address** in the selected IPsec policy.

6. From the **Protocol** list, select **Other**, and type `97` the EtherIP protocol number.
7. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
8. Click **Finished**.
   The screen refreshes and displays the new IPsec traffic selector in the list.

Implementation result

After you configure EtherIP tunneling on the BIG-IP system, you must perform the same configuration procedure on the BIG-IP system in the remote data center to fully establish the EtherIP tunnel.

After the tunnel is established, the BIG-IP system preserves any open connections to migrating (or migrated) virtual machine servers.

Chapter 5

Configuring IPsec in Tunnel Mode between Two BIG-IP Systems

Overview: Configuring IPsec between two BIG-IP systems

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. By following this procedure, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Tunnel mode (Phase 2) security associations (SAs).

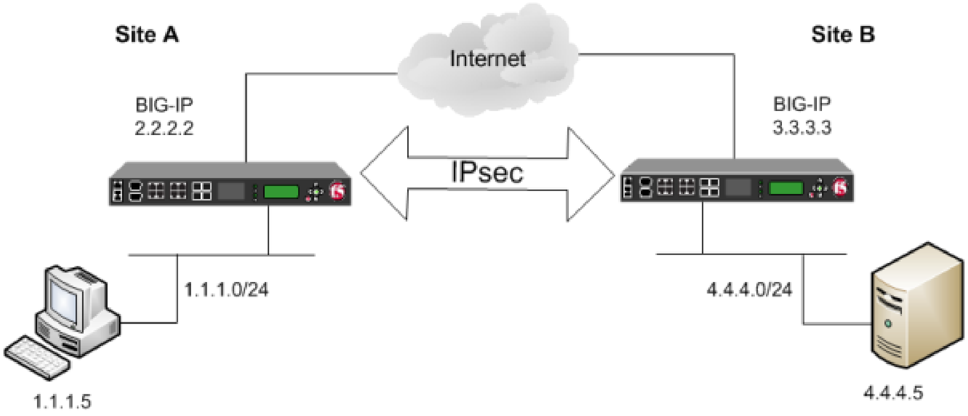


Figure 11: Example of an IPsec deployment

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers

An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system supports two versions of the IKE protocol: Version 1 (IKEv1) and Version 2 (IKEv2). The BIG-IP system includes the default IKE peer, named `anonymous`, which is configured to use Version 1.

Note: *The BIG-IP system currently supports IKEv2 only in Tunnel mode, and does not support IPComp or NAT-T with IKEv2.*

IPsec policies

An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors

A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

About IP Payload Compression Protocol (IPComp)

IP Payload Compression Protocol (IPComp) is a protocol that reduces the size of IP payloads by compressing IP datagrams before fragmenting or encrypting the traffic. IPComp is typically used to improve encryption and decryption performance, thus increasing bandwidth utilization. Using an IPsec ESP tunnel can result in packet fragmentation, because the protocol adds a significant number of bytes to a packet. The additional bytes can push the packet over the maximum size allowed on the outbound link. Using compression is one way to mitigate fragmentation. IPComp is an option when you create a custom IPsec policy.

Task summary

You can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™

This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address

Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs

These VLANs are named `external` and `internal`.

BIG-IP connectivity

Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec
Creating a custom IPsec policy
Creating a bidirectional IPsec traffic selector
Creating an IKE peer
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating a custom IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode. Another reason is to add payload compression before encryption. If you are using IKEv2, you must create a custom IPsec policy to specify in the traffic selector you create.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
7. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.

To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

Note: *Specifying a route domain other than* `0` *is supported only with IKEv2.*

This table shows sample tunnel local addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Local Address
BIG-IP A	`2.2.2.2`
BIG-IP B	`3.3.3.3`

8. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.

   This address must match the **Remote Address** setting for the relevant IKE peer. To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *Specifying a route domain other than* `0` *is supported only with IKEv2.*

   This table shows sample tunnel remote addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Remote Address
BIG-IP A	`3.3.3.3`
BIG-IP B	`2.2.2.2`

9. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
10. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
11. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
12. For the **IPComp** setting, specify whether to use IPComp encapsulation, which performs packet-level compression before encryption:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.
13. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
14. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
15. Repeat this task on the BIG-IP system in the remote location.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.

The New Traffic Selector screen opens.

3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**Last**).

   If traffic can be matched to multiple selectors, this setting specifies the priority. Traffic is matched to the traffic selector with the highest priority (lowest number).
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, type an IP address.

   This IP address should be the host or network address from which the application traffic originates. To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *Specifying a route domain other than* `0` *is supported only in IKEv2.*

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined. To specify a route domain ID in an IP address, use the format `n.n.n.n%ID`.

   Note: *Specifying a route domain other than* `0` *is supported only in IKEv2.*

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.
13. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
14. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.
15. Repeat this task on the BIG-IP system in the remote location.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other BIG-IP system with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must perform this task on both BIG-IP systems.*

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the BIG-IP system that is remote to the system you are configuring.

   To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *Specifying a route domain other than `0` is supported only with IKEv2.*

6. For the **State** setting, retain the default value, **Enabled**.
7. For the **Version** setting, select either version or both versions.

   To successfully create an IPsec tunnel, the remote IKE peer must use the same version.

   Note: *Currently, IKEv2 is supported only for Tunnel mode, which you specify when you create the IPsec policy. Some parameters are supported only by IKEv1, as indicated on the IKE Peer screens.*

   If you select both versions

   - And the system you are configuring is the IPsec initiator, the system tries using IKEv2 for negotiation. If the remote peer does not support IKEv2, the IPsec tunnel fails. To use IKEv1 in this case, clear the **Version 2** check box, and try again.
   - And the system you are configuring is the IPsec responder, the IPsec initiator system determines which IKE version to use.

8. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.
9. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.

   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*

     Note: *This option is available only for IKEv1.*

   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

Note: *The key you type must be the same at both ends of the tunnel.*

---

10. If you selected **Version 2**, select a traffic selector from the **Traffic Selector** list in the Common Settings area.

    Only traffic selectors that are valid for IKEv2 appear on the list. The default traffic selector is not included, because it is not supported in IKEv2. Also, you can associate a traffic selector with only one IKE peer, so traffic selectors already associated with other peers are not displayed.
11. If you selected **Version 2**, select **Override** from the **Presented ID** list, and enter a value in the **Presented ID Value** field.

    This value must match the **Verified ID Value** field on the remote IKE peer.
12. If you selected **Version 2**, select **Override** from the **Verified ID** list, and enter a value in the **Verified ID Value** field.

    This value must match the **Presented ID Value** field on the remote IKE peer.
13. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.
14. Repeat this task on the BIG-IP system in the remote location.

You now have an IKE peer defined for establishing a secure channel.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

---

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   ```
   tmsh modify net ipsec ike-daemon ikedaemon log-level info
   ```

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. For an IKEv1 configuration, check the IKE Phase 1 negotiation status by typing this command at the prompt.

   ```
   racoonctl -l show-sa isakmp
   ```

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent

Column	Displayed	Description
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. For an IKEv1 configuration, check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination           Status         Side
10.100.20.3       165.160.15.20         sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added

Column	Displayed
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
-----------------------------------------------------------------------------
  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                        Create:  12/13/2012:10:39
```

8. To display the details of the IKE-negotiated SAs (IKEv2), type this command at the prompt.

   `tmsh show net ipsec ike-sa all-properties`

9. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

10. Check the IPsec stats by typing this command at the prompt.

    `tmsh show net ipsec-stat`

    If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

11. If the SAs are established, but traffic is not passing, type one of these commands at the prompt.

`tmsh delete net ipsec ipsec-sa` (IKEv1)
`tmsh delete net ipsec ike-sa` (IKEv2)

This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

12. If traffic is still not passing, type this command at the prompt.

`racoonctl flush-sa isakmp`

This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

13. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established 10.100.20.3[500]-165.160.15.20[500]
spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 165.160.15.20[0]->10.100.20.3[0]
 spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 10.100.20.3[0]->165.160.15.20[0]
 spi=4573766(0x45ca46
```

14. To turn on IKEv2 logging on a production build, complete these steps.

a) Configure the log publisher for IPsec to use.

```
% tmsh create sys log-config publisher ipsec { destinations add { local-syslog }}
% tmsh list sys log-config publisher ipsec
sys log-config publisher ipsec {
    destinations {
        local-syslog { }
    }
}
```

b) Attach the log publisher to the `ike-daemon` object.

`tmsh modify net ipsec ike-daemon ikedaemon log-publisher ipsec`

15. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

`tmsh modify net ipsec ike-daemon ikedaemon log-level debug2`

Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

Note: *Using this command flushes existing SAs.*

16. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

    ```
    tmsh modify net ipsec ike-daemon ikedaemon log-level info
    ```

    Note: *Using this command flushes existing SAs.*

Implementation result

You now have an IPsec tunnel for securing traffic that traverses the WAN, from one BIG-IP® system to another.

Chapter 6

Configuring IPsec in Transport Mode between Two BIG-IP Systems

Overview: Configuring IPsec in Transport mode between two BIG-IP systems

You can configure IPsec when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP® system to another. By following this procedure, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Transport mode (Phase 2) security associations (SAs).

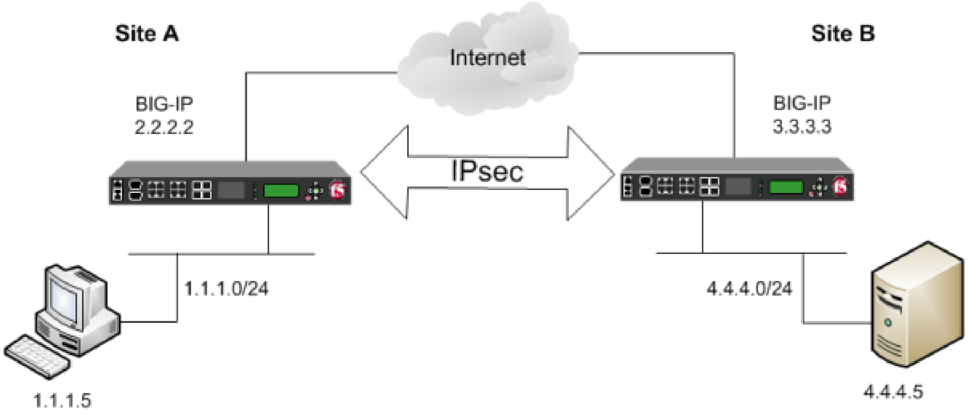


Figure 12: Example of an IPsec deployment

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Transport mode

Transport mode causes the IPsec protocol to encrypt only the payload of an IP packet. The protocol then encloses the encrypted payload in a normal IP packet. Traffic sent in Transport mode is less secure than traffic sent in Tunnel mode, because the IP header in each packet is not encrypted.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers

An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system supports two versions of the IKE protocol: Version 1 (IKEv1) and Version 2 (IKEv2). The BIG-IP system includes the default IKE peer, named `anonymous`, which is configured to use Version 1.

Note: *The BIG-IP system currently supports IKEv2 only in Tunnel mode, and does not support IPComp or NAT-T with IKEv2.*

IPsec policies

An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors

A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

About IP Payload Compression Protocol (IPComp)

IP Payload Compression Protocol (IPComp) is a protocol that reduces the size of IP payloads by compressing IP datagrams before fragmenting or encrypting the traffic. IPComp is typically used to improve encryption and decryption performance, thus increasing bandwidth utilization. Using an IPsec ESP tunnel can result in packet fragmentation, because the protocol adds a significant number of bytes to a packet. The additional bytes can push the packet over the maximum size allowed on the outbound link. Using compression is one way to mitigate fragmentation. IPComp is an option when you create a custom IPsec policy.

Task summary

With this task, you can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™

This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address

Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs

These VLANs are named `external` and `internal`.

BIG-IP connectivity

Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec
Creating an IKE peer
Creating a bidirectional IPsec policy
Creating a bidirectional IPsec traffic selector
Verifying IPsec connectivity for Transport mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other BIG-IP system with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must perform this task on both BIG-IP systems.*

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the BIG-IP system that is remote to the system you are configuring.

   To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *Specifying a route domain other than `0` is supported only with IKEv2.*

6. For the **State** setting, retain the default value, **Enabled**.
7. For the **Version** setting, select either version or both versions.

   To successfully create an IPsec tunnel, the remote IKE peer must use the same version.

   Note: *Currently, IKEv2 is supported only for Tunnel mode, which you specify when you create the IPsec policy. Some parameters are supported only by IKEv1, as indicated on the IKE Peer screens.*

   If you select both versions

   - And the system you are configuring is the IPsec initiator, the system tries using IKEv2 for negotiation. If the remote peer does not support IKEv2, the IPsec tunnel fails. To use IKEv1 in this case, clear the **Version 2** check box, and try again.
   - And the system you are configuring is the IPsec responder, the IPsec initiator system determines which IKE version to use.
8. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.
9. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.

   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*

     Note: *This option is available only for IKEv1.*
   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   Note: *The key you type must be the same at both ends of the tunnel.*
10. If you selected **Version 2**, select a traffic selector from the **Traffic Selector** list in the Common Settings area.

    Only traffic selectors that are valid for IKEv2 appear on the list. The default traffic selector is not included, because it is not supported in IKEv2. Also, you can associate a traffic selector with only one IKE peer, so traffic selectors already associated with other peers are not displayed.
11. If you selected **Version 2**, select **Override** from the **Presented ID** list, and enter a value in the **Presented ID Value** field.

    This value must match the **Verified ID Value** field on the remote IKE peer.
12. If you selected **Version 2**, select **Override** from the **Verified ID** list, and enter a value in the **Verified ID Value** field.

    This value must match the **Presented ID Value** field on the remote IKE peer.
13. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.
14. Repeat this task on the BIG-IP system in the remote location.

You now have an IKE peer defined for establishing a secure channel.

Creating a bidirectional IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode. Another reason is to add payload compression before encryption. If you are using IKEv2, you must create a custom IPsec policy to specify in the traffic selector you create.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Transport**.
7. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
8. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
9. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
10. For the **IPComp** setting, specify whether to use IPComp encapsulation, which performs packet-level compression before encryption:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.
11. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
12. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
13. Repeat this task on the BIG-IP system in the remote location.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**Last**).

If traffic can be matched to multiple selectors, this setting specifies the priority. Traffic is matched to the traffic selector with the highest priority (lowest number).

6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, type an IP address.

   This IP address should be the host or network address from which the application traffic originates. To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *Specifying a route domain other than* `0` *is supported only in IKEv2.*

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined. To specify a route domain ID in an IP address, use the format `n.n.n.n%ID`.

   Note: *Specifying a route domain other than* `0` *is supported only in IKEv2.*

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.

12. From the **Direction** list, select **Both**.
13. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
14. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.
15. Repeat this task on the BIG-IP system in the remote location.

Verifying IPsec connectivity for Transport mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the **Destination IP Address** in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.

   `racoonctl -l show-sa isakmp`

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

   This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination           Status         Side
10.100.20.3       165.160.15.20         sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3  ->  165.160.15.20  SPI(0x164208ae)  out  esp  (tmm: 0)
165.160.15.20  ->  10.100.20.3  SPI(0xfa2ca7a8)  in   esp  (tmm: 0)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20
---------------------------------------------------------------------------------------------

  tmm: 0
  Direction: out;  SPI: 0x164208ae(373426350);  Policy ID: 0x87e9(34793)
  Protocol: esp;  Mode: transport;  State: mature
  Authenticated Encryption : aes-gcm128
  Current Usage: 196 bytes
  Hard lifetime: 51 seconds; unlimited bytes
  Soft lifetime: 39 seconds; unlimited bytes
  Replay window size: 64
  Last use: 01/24/2014:14:03                                                Create:
01/24/2014:14:03

165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------------------------
```

```
  tmm: 0
  Direction: in;  SPI: 0xfa2ca7a8(4197230504);  Policy ID: 0x87e8(34792)
  Protocol: esp;  Mode: transport;  State: mature
  Authenticated Encryption : aes-gcm128
  Current Usage: 264 bytes
  Hard lifetime: 51 seconds; unlimited bytes
  Soft lifetime: 39 seconds; unlimited bytes
  Replay window size: 64
  Last use: 01/24/2014:14:03                                              Create:
01/24/2014:14:03
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that are associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3  ->  165.160.15.20  SPI(0x164208ae)  out  esp  (tmm: 0)
165.160.15.20  ->  10.100.20.3  SPI(0xfa2ca7a8)  in   esp  (tmm: 0)
```

9. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
------------------------------------------------------------------
Net::Ipsec
Cmd Id          Mode  Packets In  Bytes In  Packets Out  Bytes Out
------------------------------------------------------------------
0          TRANSPORT      353.9K    252.4M        24.9K       1.8M
0          TRANSPORT      117.9K     41.0M       163.3K      12.4M
0             TUNNEL           0         0            0          0
0             TUNNEL           0         0            0          0
1             TUNNEL           0         0            0          0
2             TUNNEL           0         0            0          0
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    `tmsh delete net ipsec ipsec-sa`

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

11. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established 10.100.20.3[500]-165.160.15.20[500]
spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Transport 165.160.15.20[0]->10.100.20.3[0]
```

```
 spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Transport 10.100.20.3[0]->165.160.15.20[0]
 spi=4573766(0x45ca46
```

13. For troubleshooting, increase the debug level by typing this command at the prompt.

    ```
    tmsh modify net ipsec ike-daemon ikedaemon log-level debug2
    ```

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

    Note: *Using this command flushes existing SAs.*

14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

    ```
    tmsh modify net ipsec ike-daemon ikedaemon log-level info
    ```

    Note: *Using this command flushes existing SAs.*

Implementation result

You now have a secure IPsec channel for securing traffic that traverses the WAN, from one BIG-IP® system to another.

Chapter 7

Configuring IPsec in Interface Mode between Two BIG-IP Systems

Overview: Configuring IPsec in Interface mode between two BIG-IP systems

You can configure an IPsec tunnel when you want to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. By following this procedure, you can create an IPsec tunnel interface that can be used as any other BIG-IP VLAN. When you configure an IPsec tunnel interface, the IKE tunnel mode security associations occur automatically as part of the tunnel negotiation. For the IPsec tunnel interface, only the IPsec Encapsulating Security Protocol (ESP) is supported for the tunnel interface, and IPComp is not available.

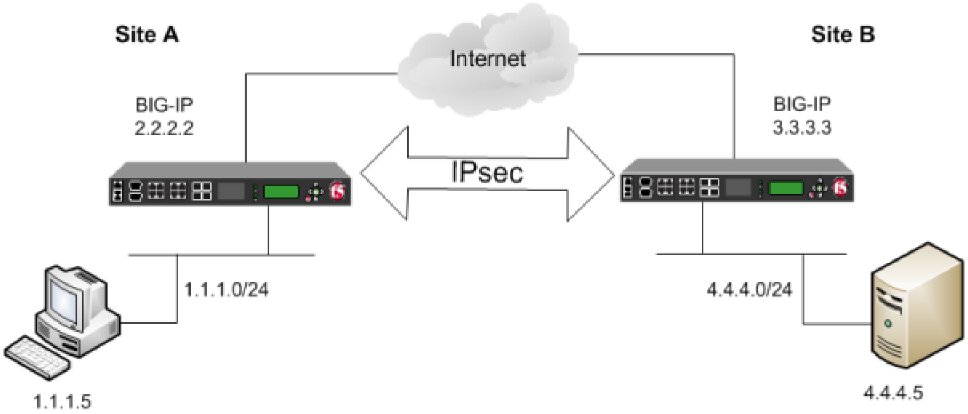


Figure 13: Example of an IPsec deployment

Task summary

Before you begin configuring IPsec, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity
Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use `ping` to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec
Creating a custom IPsec policy for Interface mode
Creating an IPsec traffic selector
Specifying an IPsec tunnel interface traffic selector
Creating an IPsec interface tunnel

Assigning a self IP address to an IP tunnel endpoint

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating a custom IPsec policy for Interface mode

You can create a custom IPsec policy to specify the Interface mode, which allows you to use the IPsec tunnel as a network interface object.

Important: *You must perform this task on the BIG-IP® systems at both sides of the tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
5. From the **Mode** list, select **IPsec Interface**.
6. Click **Finished**.
   The screen refreshes and displays the new IPsec policy in the list.
7. Repeat this task on the BIG-IP system in the remote location.

Creating an IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses you specify and the custom IPsec policy you assign.

Important: *You must perform this task on the BIG-IP® systems on both sides of the WAN.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.

3. In the **Name** field, type a unique name for the traffic selector.
4. For the **Source IP Address** setting, specify where the application traffic originates, either:
   - Click **Host** and type an IP address.
   - Click **Network**, and in the **Address** field, type an IP address.

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

5. For the **Destination IP Address** setting, specify where the application traffic is going, either:
   - Click **Host** and type an IP address.
   - Click **Network**, and in the **Address** field, type an IP address.

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

6. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
7. Click **Finished**.
   The screen refreshes and displays the new IPsec traffic selector in the list.
8. Repeat this task on the BIG-IP system in the remote location.

Specifying an IPsec tunnel interface traffic selector

You can create an IPsec tunnel profile to filter traffic according to the traffic selector you specify.

1. On the Main tab, click **Network** > **Tunnels** > **Profiles** > **IPsec** > **Create**.
   The New IPsec Profile screen opens.
2. In the **Name** field, type a unique name for the profile.
3. From the **Parent Profile** list, select **ipsec**.
4. Select the **Custom** check box.
5. From the **Traffic Selector** list, select the traffic selector you created.
6. Click **Finished**.

To use this IPsec profile to filter traffic, you must apply it to an IPsec tunnel.

Creating an IPsec interface tunnel

You can create an IPsec interface tunnel to apply an IPsec profile you have created to specify the traffic selector to filter the traffic.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.

2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **IPsec**.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, select **Specify**, and type the IP address of the BIG-IP device at the other end of the tunnel.
6. Click **Finished**.

After you create an IPsec tunnel interface, you can use it just like any other tunnel interface, such as assigning it a self IP address, associating it with route domains, and adding it to virtual servers.

Assigning a self IP address to an IP tunnel endpoint

Ensure that you have created an IP tunnel before starting this task.

Self IP addresses can enable the BIG-IP® system, and other devices on the network, to route application traffic through the associated tunnel, similar to routing through VLANs and VLAN groups.

Note: *If the other side of the tunnel needs to be reachable, make sure the self IP addresses that you assign to both sides of the tunnel are in the same subnet.*

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type the IP address of the tunnel.

   The system accepts IPv4 and IPv6 addresses.

   Note: *This is not the same as the IP address of the tunnel local endpoint.*
5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select the tunnel with which to associate this self IP address.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

Assigning a self IP to a tunnel ensures that the tunnel appears as a resource for routing traffic.

To direct traffic through the tunnel, add a route for which you specify the tunnel as the resource.

Chapter 8

Configuring IPsec between a BIG-IP System and a Third-Party Device

Overview: Configuring IPsec between a BIG-IP system and a third-party device

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from a BIG-IP® system to third-party device. By following this process, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Tunnel mode (Phase 2) security associations (SAs).

This implementation describes the tasks for setting up the IPsec tunnel on the BIG-IP system. You must also configure the third-party device at the other end of the tunnel. For those instructions, refer to the manufacturer's documentation for your device.

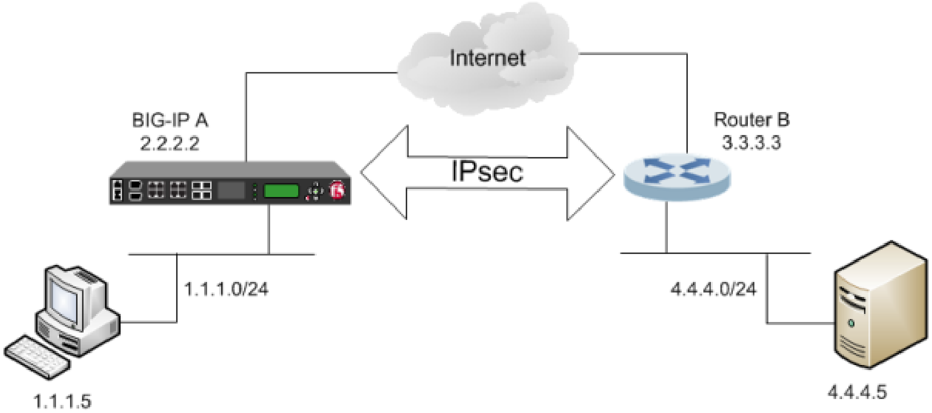


Figure 14: Example of an IPsec tunnel between a BIG-IP system and a third-party device

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers

An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system supports two versions of the IKE protocol: Version 1 (IKEv1) and Version 2 (IKEv2). The BIG-IP system includes the default IKE peer, named `anonymous`, which is configured to use Version 1.

Note: *The BIG-IP system currently supports IKEv2 only in Tunnel mode, and does not support IPComp or NAT-T with IKEv2.*

IPsec policies

An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors

A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

Task summary

You can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that this module, system objects, and connectivity exist on the BIG-IP® system:

BIG-IP Local Traffic Manager™

This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address

The BIG-IP system must have at least one self IP address, to be used in specifying the end of the IPsec tunnel.

The default VLANs

These VLANs are named `external` and `internal`.

BIG-IP connectivity

Verify the connectivity between the client or server and its BIG-IP device, and between the BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec

Creating an IKE peer
Creating a custom IPsec policy
Creating a bidirectional IPsec traffic selector
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other device with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the device that is remote to the system you are configuring.

   This address must match the value of the **Tunnel Remote Address** setting in the relevant IPsec policy.
6. For the **State** setting, retain the default value, **Enabled**.
7. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

   Important: *The values you select must match the IKE Phase 1 settings on the remote device.*

Setting	Options
Authentication Algorithm	**MD5**

Setting	Options
	SHA-1 (default) **SHA-256** **SHA-384** **SHA-512**
Encryption Algorithm	**DES** **3 DES** (default) **BLOWFISH** **CAST128** **AES** **CAMELLIA**
Perfect Forward Secrecy	**MODP768** **MODP1024** (default) **MODP1536** **MODP2048** **MODP3072** **MODP4096** **MODP6144** **MODP8192**
Lifetime	Length of time, in minutes, before the IKE security association expires.

8. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.
   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*

     Note: *This option is available only for IKEv1.*
   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   Note: *The key you type must be the same at both ends of the tunnel.*
9. For the Common Settings area, retain all default values.
10. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.

You now have an IKE peer defined for establishing a secure channel.

Creating a custom IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
7. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   For example, the tunnel local IP address for BIG-IP A is `2.2.2.2`.
8. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.

   This address must match the **Remote Address** setting for the relevant IKE peer.

   For example, the tunnel remote IP address configured on BIG-IP A is the IP address of Router B, which is `3.3.3.3`.
9. For the IKE Phase 2 area, retain the default values, or select the options that are appropriate for your deployment.

   Important: *The values you select must match the IKE Phase 2 settings on the remote device.*

Setting	Options
Authentication Algorithm	**SHA-1** **AES-GCM128** (default) **AES-GCM192** **AES-GCM256** **AES-GMAC128** **AES-GMAC192** **AES-GMAC256**
Encryption Algorithm	**AES-GCM128** (default)
Perfect Forward Secrecy	**MODP768** **MODP1024** (default) **MODP1536** **MODP2048** **MODP3072** **MODP4096** **MODP6144** **MODP8192**
Lifetime	Length of time, in minutes, before the IKE security association expires.

10. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**First**).

   This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, click **Host** or **Network**, and in the **Address** field, type an IP address.

   This IP address should be the host or network address from which the application traffic originates.

   This table shows sample source IP addresses for BIG-IP A and Router B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
Router B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, click **Host**, and in the **Address** field, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined.

   This table shows sample destination IP addresses for BIG-IP A and Router B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
Router B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.
13. From the **Action** list, select **Protect**.
    The **IPsec Policy Name** setting appears.
14. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
15. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

***Note:** Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   ```
   tmsh modify net ipsec ike-daemon ikedaemon log-level info
   ```

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. For an IKEv1 configuration, check the IKE Phase 1 negotiation status by typing this command at the prompt.

   ```
   racoonctl -l show-sa isakmp
   ```

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. For an IKEv1 configuration, check the IKE Phase 2 negotiation status by typing this command at the prompt.

   ```
   racoonctl -ll show-sa internal
   ```

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination           Status         Side
10.100.20.3       165.160.15.20         sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
-----------------------------------------------------------------------------
  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                        Create:  12/13/2012:10:39
```

8. To display the details of the IKE-negotiated SAs (IKEv2), type this command at the prompt.

   `tmsh show net ipsec ike-sa all-properties`

9. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

10. Check the IPsec stats by typing this command at the prompt.

    `tmsh show net ipsec-stat`

    If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

11. If the SAs are established, but traffic is not passing, type one of these commands at the prompt.

    `tmsh delete net ipsec ipsec-sa` (IKEv1)
    `tmsh delete net ipsec ike-sa` (IKEv2)

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

12. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

13. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established 10.100.20.3[500]-165.160.15.20[500]
spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 165.160.15.20[0]->10.100.20.3[0]
 spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 10.100.20.3[0]->165.160.15.20[0]
 spi=4573766(0x45ca46
```

14. To turn on IKEv2 logging on a production build, complete these steps.

a) Configure the log publisher for IPsec to use.

```
% tmsh create sys log-config publisher ipsec { destinations add { local-syslog }}
% tmsh list sys log-config publisher ipsec
sys log-config publisher ipsec {
    destinations {
        local-syslog { }
    }
}
```

b) Attach the log publisher to the `ike-daemon` object.

```
tmsh modify net ipsec ike-daemon ikedaemon log-publisher ipsec
```

15. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level debug2
```

Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

Note: *Using this command flushes existing SAs.*

16. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level info
```

Note: *Using this command flushes existing SAs.*

Implementation result

You now have an IPsec tunnel for securing traffic that traverses the WAN, from one BIG-IP® system to a third-party device.

Chapter

9

Configuring IPsec Using Manually Keyed Security Associations

Overview: Configuring IPsec using manually keyed security associations

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. Typically, you would use the Internet Key Exchange (IKE) protocol to negotiate the secure channel between the two systems. If you choose not to use IKE, you must create manual security associations for IPsec security. A *manual security association* statically defines the specific attribute values that IPsec should use for the authentication and encryption of data flowing through the tunnel.

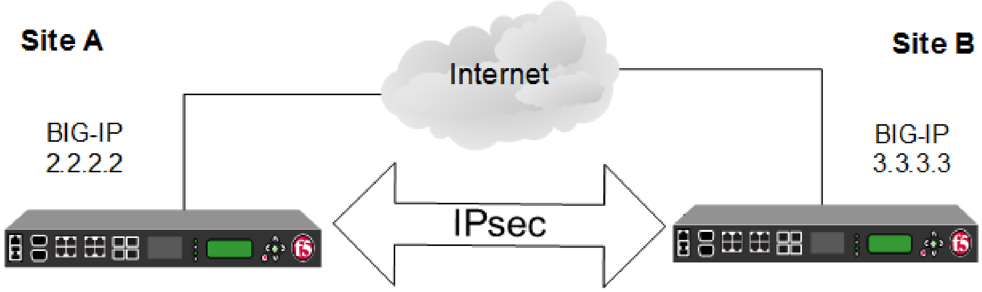


Figure 15: Illustration of an IPsec deployment

The implementation of the IPsec protocol suite with a manual security association consists of these components:

IPsec policy

An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession®). For Tunnel mode, the policy also specifies the endpoints for the tunnel. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets.

Manual security association

A *manual security association* is set of information that the IPsec protocol uses to authenticate and encrypt application traffic.

Note: *When you manually create a security association instead of using IKE, the peer systems do not negotiate these attributes. Peers can communicate only when they share the same configured attributes.*

Traffic selector

A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

Task summary

You can configure an IPsec tunnel to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP system connectivity
Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use `ping` to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec
Creating custom IPsec policies for manual security associations
Manually creating IPsec security associations for inbound and outbound traffic
Creating IPsec traffic selectors for manually keyed security associations
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating custom IPsec policies for manual security associations

When you are using manual security associations for an IPsec tunnel between two BIG-IP® systems, you must create two custom IPsec policies on each system, one to use for outbound traffic and the other for inbound traffic. You establish the directionality of a policy by associating it with a unidirectional traffic selector.

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
5. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
6. In the **Tunnel Local Address** field, type the IP address of the BIG-IP system that initiates the traffic.

   To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *When you use IKEv1, the BIG-IP system supports a maximum of 512 route domains.*

   For the outbound policy, this is the IP address of the local BIG-IP system. For the inbound policy, this is the IP address of the remote BIG-IP system.

   This table shows sample outbound and inbound tunnel local addresses configured on BIG-IP A and BIG-IP B.

System Name	Traffic Direction	Tunnel Local Address
BIG-IP A	Outbound	`2.2.2.2`
	Inbound	`3.3.3.3`
BIG-IP B	Outbound	`3.3.3.3`
	Inbound	`2.2.2.2`

7. In the **Tunnel Remote Address** field, type the IP address of the BIG-IP system that receives the traffic.

   To specify a route domain ID in an IP address, use the format n.n.n.n%ID.

   Note: *When you use IKEv1, the BIG-IP system supports a maximum of 512 route domains.*

   For the outbound policy, this is the IP address of the remote BIG-IP system. For the inbound policy, this is the IP address of the local BIG-IP system.

   This table shows sample outbound and inbound tunnel remote addresses configured on BIG-IP A and BIG-IP B.

System Name	Traffic Direction	Tunnel Remote Address
BIG-IP A	Outbound	`3.3.3.3`
	Inbound	`2.2.2.2`
BIG-IP B	Outbound	`2.2.2.2`
	Inbound	`3.3.3.3`

8. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
9. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
10. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
11. For the **IPComp** setting, specify whether to use IPComp encapsulation, which performs packet-level compression before encryption:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.
12. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
13. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
14. Repeat this task for outbound and inbound traffic policies on both the local and remote BIG-IP systems.

When you are finished, you should have created four separate IPsec policies, two on each system.

Manually creating IPsec security associations for inbound and outbound traffic

Before you start this task, you need to create two custom IPsec policies on the BIG-IP® system, one for outbound traffic and another for inbound traffic.

You can manually create security associations to specify the security attributes for a given IPsec communication session. For the manual configuration, you need to create two manual security associations for each connection, one for outbound traffic and the other for inbound traffic.

Important: *You must perform this task on both BIG-IP systems.*

1. On the Main tab, click **Network** > **IPsec** > **Manual Security Associations**.
2. Click the **Create** button.
   The New Security Association screen opens.
3. In the **Name** field, type a unique name for the security association.
4. In the **Description** field, type a brief description of the security setting.
5. In the **SPI** field, type a unique number for the security parameter index.

   This number must be an integer between 256 and 4294967296.
6. In the **Source Address** field, type the source IP address.
7. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.
8. In the **Authentication Key** field, type a key value.

   This value can by any double-quoted character string up to a maximum of 128 characters
9. From the **Encryption Algorithm** list, select the algorithm appropriate to your deployment.
10. In the **Encryption Key** field, type a key value.

    This value can by any double-quoted character string up to a maximum of 128 characters

11. From the **IPsec Policy Name** list, select an IPsec policy.
    - For the outbound security association, select the IPsec policy you created for outbound traffic.
    - For the inbound security association, select the IPsec policy you created for inbound traffic.
12. Repeat this task for security associations that handle outbound and inbound traffic on both the local and remote BIG-IP systems.

When you are finished, you should have manually created four separate security associations, two on each system.

Creating IPsec traffic selectors for manually keyed security associations

Before you start this task, you need to create two custom IPsec policies on the BIG-IP® system, one for outbound traffic and another for inbound traffic.

You can use this procedure to create IPsec traffic selectors that reference custom IPsec policies for unidirectional traffic in an IPsec tunnel for which you have manually keyed security associations. You need to create two traffic selectors on each BIG-IP system, one for outbound traffic and the other for inbound traffic. Each *traffic selector* you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. From the **Configuration** list, select **Advanced**.
6. For the **Source IP Address or CIDR** setting, type an IP address.

   This IP address must match the IP address specified for the **Tunnel Local Address** in the selected IPsec policy.
7. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
8. For the **Destination IP Address or CIDR** setting, type an IP address.

   This IP address must match the IP address specified for the **Tunnel Remote Address** in the selected IPsec policy.
9. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
10. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
11. From the **Direction** list, select **Out** or **In**, depending on whether this traffic selector is for outbound or inbound traffic.
12. From the **IPsec Policy Name** list, select an IPsec policy.
    - For the outbound traffic selector, select the IPsec policy you created for outbound traffic.
    - For the inbound traffic selector, select the IPsec policy you created for inbound traffic.
13. Click **Finished**.

The screen refreshes and displays the new IPsec traffic selector in the list.

14. Repeat this task for traffic selectors that handle outbound and inbound traffic on both the local and remote BIG-IP systems.

When you are finished, you should have manually created four separate traffic selectors, two on each system.

Verifying IPsec connectivity for Tunnel mode

After you have manually configured security associations for an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

***Note:** Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Send data traffic to the destination IP address specified in the traffic selector.
3. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

4. To verify the establishment of manually configured security associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

5. To display the details of the manually configured security associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
-----------------------------------------------------------------------------
  tmm: 2
```

```
Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
Protocol: esp;  Mode: tunnel;  State: mature
Authenticated Encryption : aes-gmac128
Current Usage: 307488 bytes
Hard lifetime: 94 seconds; unlimited bytes
Soft lifetime: 34 seconds; unlimited bytes
Replay window size: 64
Last use: 12/13/2012:10:42                        Create:  12/13/2012:10:39
```

Chapter 10

Setting Up IPsec To Use NAT Traversal on Both Sides of the WAN

Overview: Setting up IPsec to use NAT traversal on both sides of the WAN

When you are using IPsec to secure WAN traffic, you can set up an IPsec tunnel with NAT traversal (NAT-T) to get around a firewall or other NAT device. This implementation describes how to set up the IPsec tunnel when you have a NAT device on both sides of the tunnel.

The following illustration shows a network configuration with a firewall on both sides of the WAN.

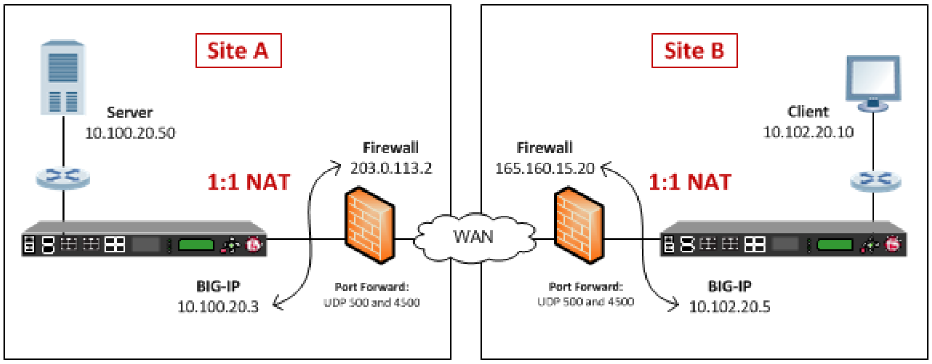


Figure 16: Example of an IPsec deployment with NAT-T on both sides of the WAN

Before you begin IPsec configuration

Before you configure IPsec on a BIG-IP® device, make sure that you have completed the following general prerequisites.

- You must have an existing routed IP network between the two locations where the BIG-IP devices will be installed.
- The BIG-IP hardware is installed with an initial network configuration applied.
- The management IP address is configured on the BIG-IP system.
- If you are using NAT traversal, forward UDP ports 500 and 4500 to the BIG-IP system behind each firewall.
- Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. You can use ping to test connectivity.

Task summary

When you are configuring an IPsec tunnel, you must repeat the configuration tasks on the BIG-IP systems on both sides of the WAN.

Creating a forwarding virtual server for IPsec
Creating an IPsec tunnel with NAT-T on both sides
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IPsec tunnel with NAT-T on both sides

You can create an IPsec tunnel to securely transport application traffic across the WAN. You must configure the IPsec tunnel on the BIG-IP systems on both sides of the WAN.

When you create an IKE peer for NAT traversal (NAT-T), the key configuration detail is that the **Remote Address** setting is the public IP address of the firewall or other NAT device (not the IP address of the remote BIG-IP system). Also, you must turn on NAT traversal. You can customize the remaining settings to conform to your network.

Important: *For the IKE peer negotiations to be successful, the IKE Phase 1 and IKE Phase 2 settings must be the same on the BIG-IP systems at both ends of the IPsec tunnel.*

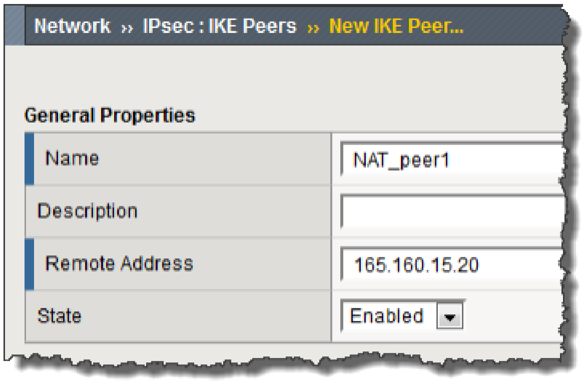
1. Create an IKE peer that specifies the other end of the IPsec tunnel.
   a) On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
   b) Click the **Create** button.
   c) In the **Name** field, type a unique name for the IKE peer.
   d) In the **Remote Address** field, type the public IP address of the firewall or other NAT device that is between the WAN and the remote BIG-IP system.

      This address is the IP address of the remote peer, and must match the value of the **Tunnel Remote Address** setting in the relevant IPsec policy.

      For example, the peer remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Remote (Peer) Address
Site A	`165.160.15.20`
Site B	`203.0.113.2`

This screen snippet shows the peer **Remote Address** setting at Site A.



Network » IPsec : IKE Peers » New IKE Peer...

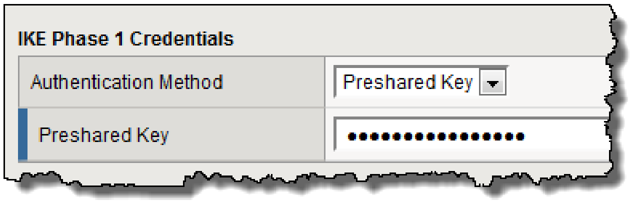
General Properties

Name	NAT_peer1
Description	
Remote Address	165.160.15.20
State	Enabled

e) For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

f) In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **Preshared Key** or **RSA Signature**, and specify additional information in the fields that appear.

For example, if you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

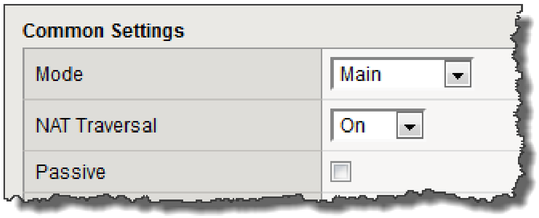


IKE Phase 1 Credentials

Authentication Method	Preshared Key
Preshared Key	••••••••••••••••

Note: *The key you type must be the same at both ends of the tunnel.*

g) From the **NAT Traversal** list, select **On**.



Common Settings

Mode	Main
NAT Traversal	On
Passive	□

h) Click **Finished**.

2. Create a custom IPsec policy that uses Tunnel mode and has the same remote IP address as the IKE peer.

   a) On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.

   b) Click the **Create** button.

   c) In the **Name** field, type a unique name for the policy.

   d) For the **IPsec Protocol** setting, retain the default selection, **ESP**.

   e) From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.

   f) In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   For example, the tunnel local addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Local Address
Site A	`10.100.20.3`
Site B	`10.102.20.5`

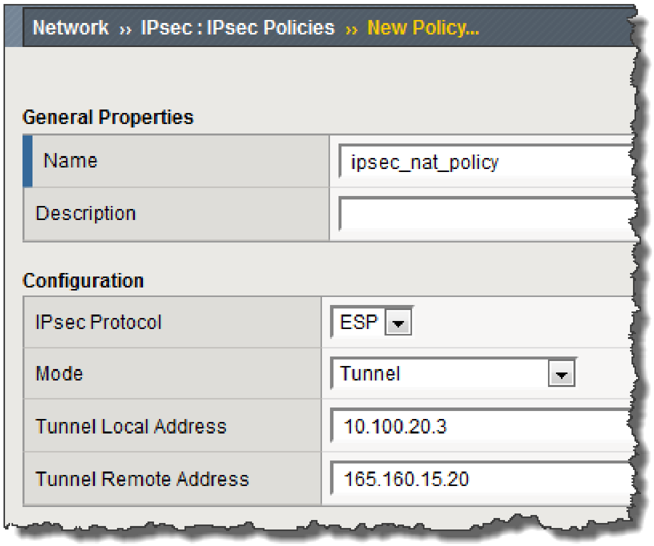
g) In the **Tunnel Remote Address** field, type the public IP address of the firewall or other NAT device that is between the WAN and the remote BIG-IP system.

This address must match the value of the **Remote Address** setting for the relevant IKE peer.

For example, the tunnel remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Remote Address
Site A	`165.160.15.20`
Site B	`203.0.113.2`

This screen snippet shows the tunnel settings at Site A.



General Properties

Name	ipsec_nat_policy
Description	

Configuration

IPsec Protocol	ESP
Mode	Tunnel
Tunnel Local Address	10.100.20.3
Tunnel Remote Address	165.160.15.20

h) For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.

i) For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.

j) For the **Perfect Forward Secrecy** setting, retain the default value, or select the option appropriate for your deployment.

k) Click **Finished**.

3. Create a bidirectional traffic selector that uses the custom IPsec policy you created.

The traffic selector filters the application traffic based on the source and destination IP addresses you specify.

a) On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.

b) Click **Create**.

c) In the **Name** field, type a unique name for the traffic selector.

d) For the **Order** setting, retain the default value (**First**).

e) For the **Source IP Address** setting, in the **Address** field, type the IP address from which the application traffic originates.
For example, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

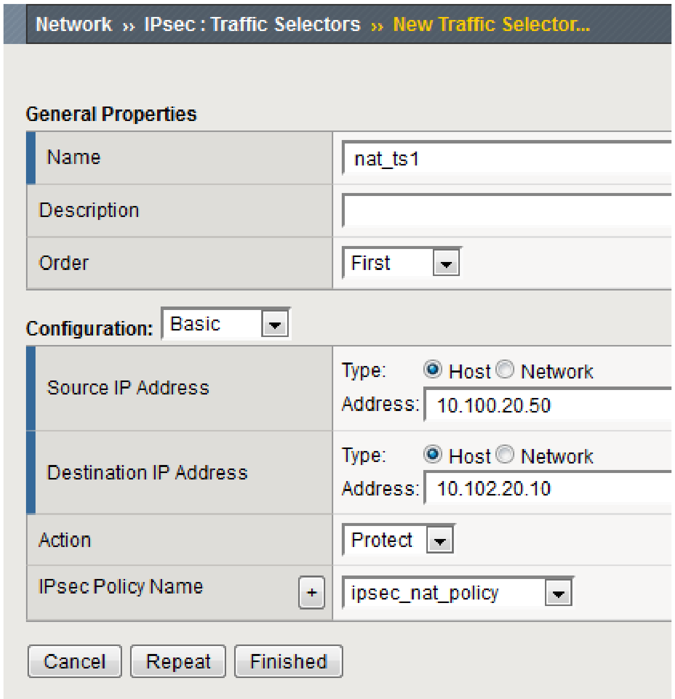
Location	Source IP Address
Site A	`10.100.20.50`
Site B	`10.102.20.10`

f) In the **Destination IP Address** setting **Address** field, type the final IP address for which the application traffic is destined.
For example, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Destination IP Address
Site A	`10.102.20.10`
Site B	`10.100.20.50`

g) For the **Action** setting, retain the default value, **Protect**.

h) From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you just created.
This portion of a screen is an example of the completed Traffic Selector screen at Site A.


Network » IPsec : Traffic Selectors » New Traffic Selector...

General Properties

Name	nat_ts1
Description	
Order	First

Configuration: Basic

Source IP Address	Type: ● Host ○ Network Address: 10.100.20.50
Destination IP Address	Type: ● Host ○ Network Address: 10.102.20.10
Action	Protect
IPsec Policy Name	ipsec_nat_policy

i) Click **Finished**.

You have now created an IPsec tunnel through which traffic travels in both directions across the WAN through firewalls on both sides.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   ```
   tmsh modify net ipsec ike-daemon ikedaemon log-level info
   ```

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. For an IKEv1 configuration, check the IKE Phase 1 negotiation status by typing this command at the prompt.

   ```
   racoonctl -l show-sa isakmp
   ```

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. For an IKEv1 configuration, check the IKE Phase 2 negotiation status by typing this command at the prompt.

   ```
   racoonctl -ll show-sa internal
   ```

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination           Status         Side
10.100.20.3       165.160.15.20         sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   ```
   tmsh show net ipsec ipsec-sa
   ```

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   ```
   tmsh show net ipsec ipsec-sa all-properties
   ```

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
-----------------------------------------------------------------------------
  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                        Create:  12/13/2012:10:39
```

8. To display the details of the IKE-negotiated SAs (IKEv2), type this command at the prompt.

   `tmsh show net ipsec ike-sa all-properties`

9. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

10. Check the IPsec stats by typing this command at the prompt.

    `tmsh show net ipsec-stat`

    If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

11. If the SAs are established, but traffic is not passing, type one of these commands at the prompt.

    `tmsh delete net ipsec ipsec-sa` (IKEv1)
    `tmsh delete net ipsec ike-sa` (IKEv2)

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

12. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

13. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established 10.100.20.3[500]-165.160.15.20[500]
spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 165.160.15.20[0]->10.100.20.3[0]
 spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 10.100.20.3[0]->165.160.15.20[0]
 spi=4573766(0x45ca46
```

14. To turn on IKEv2 logging on a production build, complete these steps.

    a) Configure the log publisher for IPsec to use.

```
% tmsh create sys log-config publisher ipsec { destinations add { local-syslog }}
% tmsh list sys log-config publisher ipsec
sys log-config publisher ipsec {
    destinations {
        local-syslog { }
    }
}
```

b) Attach the log publisher to the `ike-daemon` object.

```
tmsh modify net ipsec ike-daemon ikedaemon log-publisher ipsec
```

15. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level debug2
```

Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

Note: *Using this command flushes existing SAs.*

16. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level info
```

Note: *Using this command flushes existing SAs.*

Chapter 11

Setting Up IPsec To Use NAT Traversal on One Side of the WAN

Overview: Setting up IPsec to use NAT traversal on one side of the WAN

When you are using IPsec to secure WAN traffic, you can set up an IPsec tunnel with NAT traversal (NAT-T) to get around a firewall or other NAT device. This implementation describes how to set up the IPsec tunnel when you have a NAT device on one side of the tunnel.

The following illustration shows a network configuration with a firewall on one side of the WAN.

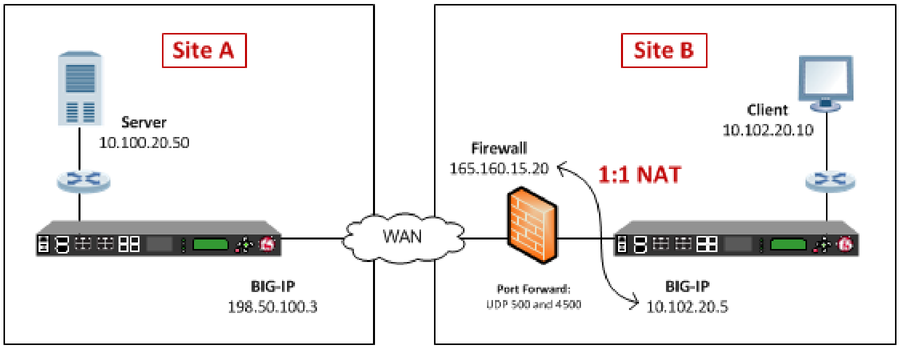


Figure 17: Example of an IPsec deployment with NAT-T on one side of the WAN

Before you begin IPsec configuration

Before you configure IPsec on a BIG-IP® device, make sure that you have completed the following general prerequisites.

- You must have an existing routed IP network between the two locations where the BIG-IP devices will be installed.
- The BIG-IP hardware is installed with an initial network configuration applied.
- The management IP address is configured on the BIG-IP system.
- If you are using NAT traversal, forward UDP ports 500 and 4500 to the BIG-IP system behind each firewall.
- Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. You can use ping to test connectivity.

Task summary

When you are configuring an IPsec tunnel, you must repeat the configuration tasks on the BIG-IP systems on both sides of the WAN.

Creating a forwarding virtual server for IPsec
Creating an IPsec tunnel with NAT-T on one side
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. In the **Destination Address** field, type a wildcard network address in CIDR format, such as `0.0.0.0/0` for IPv4 or `::/0` for IPv6, to accept any traffic.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IPsec tunnel with NAT-T on one side

You can create an IPsec tunnel to securely transport application traffic across the WAN. You must configure an IPsec tunnel on the BIG-IP systems on both sides of the WAN.

When you create an IKE peer for NAT traversal (NAT-T), the key configuration detail is that the **Remote Address** setting is the public IP address of the firewall or other NAT device (not the IP address of the remote BIG-IP system). Also, you must turn on NAT traversal for that peer. You can customize the remaining settings to conform to your network.

Important: *For the IKE peer negotiations to be successful, the IKE Phase 1 and IKE Phase 2 settings must be the same on the BIG-IP systems at both ends of the IPsec tunnel.*

1. Create an IKE peer that specifies the other end of the IPsec tunnel.
   a) On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
   b) Click the **Create** button.
   c) In the **Name** field, type a unique name for the IKE peer.
   d) In the **Remote Address** field, type the IP address of the remote peer.

   If the remote BIG-IP system is behind a firewall or other NAT device, type the public IP address of that device.

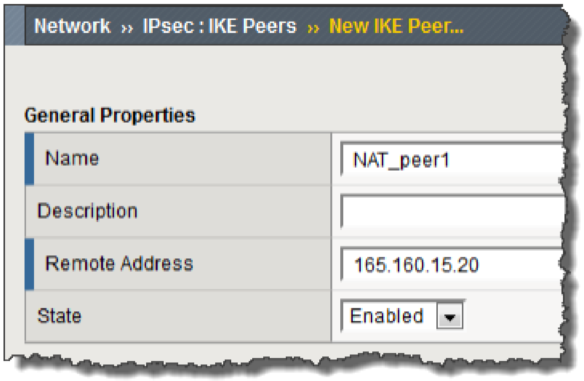
   If the remote BIG-IP system is reachable directly, type the IP address of the BIG-IP system.

   Note: *This address must match the value of the* ***Tunnel Remote Address*** *of the remote site setting in the relevant IPsec policy.*

   For example, Site A uses the WAN IP address of the Site B firewall. The peer remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Remote (Peer) Address
Site A	`165.160.15.20`
Site B	`198.50.100.3`

This screen snippet shows the peer **Remote Address** setting at Site A.



Network » IPsec : IKE Peers » New IKE Peer...

General Properties

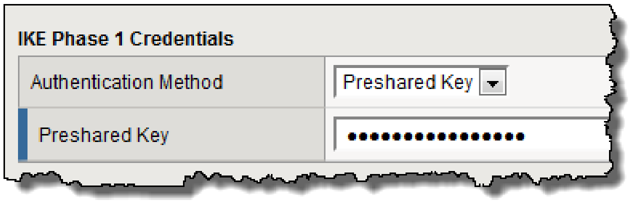
Name	NAT_peer1
Description	
Remote Address	165.160.15.20
State	Enabled

e) For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

f) For the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **Preshared Key** or **RSA Signature**, and specify additional information in the fields that appear.

For example, if you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

In this example, **Preshared Key** is selected.



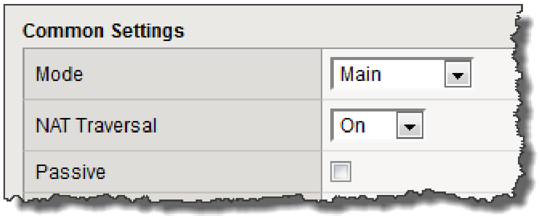
IKE Phase 1 Credentials

Authentication Method	Preshared Key
Preshared Key	••••••••••••••••

Note: *The key you type must be the same at both ends of the tunnel.*

g) From the **NAT Traversal** list, select **On** for Site A's IKE peer.

Note: *Use this setting only for the IKE peer (remote BIG-IP system) that is behind a NAT device. On the Site B BIG-IP system, for the IKE peer, retain the default setting,* ***Off****.*



Common Settings

Mode	Main
NAT Traversal	On
Passive	□

h) Click **Finished**.

2. Create a custom IPsec policy that uses Tunnel mode and has the same remote IP address as the IKE peer.

a) On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.

b) Click the **Create** button.

c) In the **Name** field, type a unique name for the policy.

d) For the **IPsec Protocol** setting, retain the default selection, **ESP**.

e) From the **Mode** list, select **Tunnel**.
The screen refreshes to show additional related settings.

f) In the **Tunnel Local Address** field, type the local IP address of the system you are configuring. For example, the tunnel local addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Local Address
Site A	`198.50.100.3`
Site B	`10.102.20.5`

g) In the **Tunnel Remote Address** field, type the IP address of the remote peer.

If the remote BIG-IP system is behind a firewall or other NAT device, type the public IP address of that device.

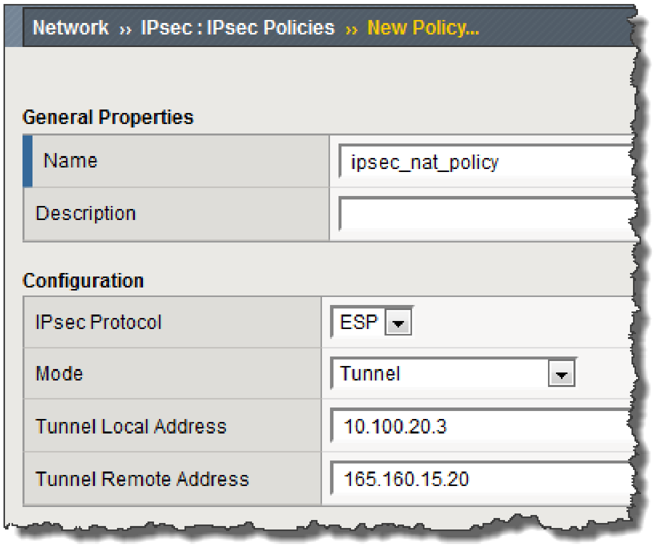
If the remote BIG-IP system is reachable directly, type the IP address of the BIG-IP system.

Note: *This address must match the value of the* ***Remote Address*** *setting in the relevant IKE peer.*

For example, the tunnel remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Remote Address
Site A	`165.160.15.20`
Site B	`198.50.100.3`

This screen snippet shows the tunnel settings at Site A.



General Properties

Name	ipsec_nat_policy
Description	

Configuration

IPsec Protocol	ESP
Mode	Tunnel
Tunnel Local Address	10.100.20.3
Tunnel Remote Address	165.160.15.20

h) For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.

i) For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.

j) For the **Perfect Forward Secrecy** setting, retain the default value, or select the option appropriate for your deployment.

k) Click **Finished**.

3. Create a bidirectional traffic selector that uses the custom IPsec policy you created.

The traffic selector filters the application traffic based on the source and destination IP addresses you specify.

a) On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.

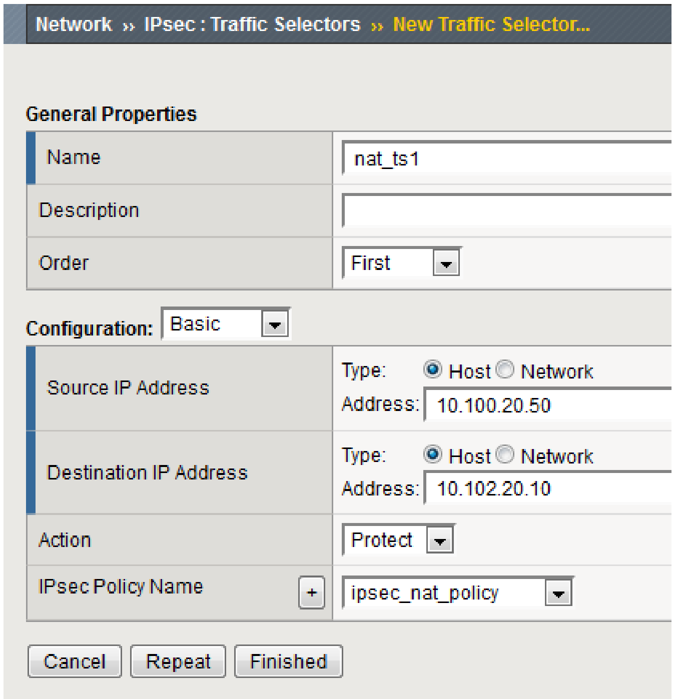
b) Click **Create**.
c) In the **Name** field, type a unique name for the traffic selector.
d) For the **Order** setting, retain the default value (**First**).
e) For the **Source IP Address** setting, in the **Address** field, type the IP address from which the application traffic originates.
In the illustration the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Source IP Address
Site A	`10.100.20.50`
Site B	`10.102.20.10`

f) For the **Destination IP Address** setting, in the **Address** field, type the final IP address for which the application traffic is destined.
In the illustration, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Destination IP Address
Site A	`10.102.20.10`
Site B	`10.100.20.50`

g) For the **Action** setting, retain the default value, **Protect**.
h) From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you just created.
This screen snippet is an example of the completed Traffic Selector screen at Site A.


Network ›› IPsec : Traffic Selectors ›› New Traffic Selector...
General Properties
Configuration: Basic
Cancel
Repeat
Finished

Name	nat_ts1
Description	
Order	First
Source IP Address	Type: ● Host ○ Network Address: 10.100.20.50
Destination IP Address	Type: ● Host ○ Network Address: 10.102.20.10
Action	Protect
IPsec Policy Name	ipsec_nat_policy

i) Click **Finished**.

You have now created an IPsec tunnel through which traffic travels in both directions across the WAN, and through a firewall on one side.

Task summary

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   ```
   tmsh modify net ipsec ike-daemon ikedaemon log-level info
   ```

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. For an IKEv1 configuration, check the IKE Phase 1 negotiation status by typing this command at the prompt.

   ```
   racoonctl -l show-sa isakmp
   ```

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. For an IKEv1 configuration, check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination            Status         Side
10.100.20.3       165.160.15.20          sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
```

```
165.160.15.20 -> 10.100.20.3
-----------------------------------------------------------------------------
  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                        Create:  12/13/2012:10:39
```

8. To display the details of the IKE-negotiated SAs (IKEv2), type this command at the prompt.

   ```
   tmsh show net ipsec ike-sa all-properties
   ```

9. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   ```
   tmsh show net ipsec ipsec-sa traffic-selector ts_codec
   ```

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

10. Check the IPsec stats by typing this command at the prompt.

    ```
    tmsh show net ipsec-stat
    ```

    If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

11. If the SAs are established, but traffic is not passing, type one of these commands at the prompt.

    `tmsh delete net ipsec ipsec-sa` (IKEv1)
    `tmsh delete net ipsec ike-sa` (IKEv2)

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

12. If traffic is still not passing, type this command at the prompt.

    ```
    racoonctl flush-sa isakmp
    ```

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

13. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established 10.100.20.3[500]-165.160.15.20[500]
spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 165.160.15.20[0]->10.100.20.3[0]
 spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel 10.100.20.3[0]->165.160.15.20[0]
 spi=4573766(0x45ca46
```

14. To turn on IKEv2 logging on a production build, complete these steps.

a) Configure the log publisher for IPsec to use.

```
% tmsh create sys log-config publisher ipsec { destinations add { local-syslog }}
% tmsh list sys log-config publisher ipsec
sys log-config publisher ipsec {
    destinations {
        local-syslog { }
    }
}
```

b) Attach the log publisher to the `ike-daemon` object.

```
tmsh modify net ipsec ike-daemon ikedaemon log-publisher ipsec
```

15. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level debug2
```

Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

Note: *Using this command flushes existing SAs.*

16. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level info
```

Note: *Using this command flushes existing SAs.*

Chapter 12

Diagnosing IPsec Tunnel Issues

Overview: Diagnosing IPsec tunnel issues

Using the browser interface, you can diagnose problems with the IPsec tunnels you create on the BIG-IP® system. The IPsec diagnostics search capability facilitates quick retrieval of data, even when you have a large number of IPsec tunnels. The search results list the traffic selector that meets your criteria. You can search on source IP address, destination IP address, both source and destination IP addresses, IPsec policy name, or traffic selector name.

To search on the source or destination IP address of a traffic selector, you can type either a valid IPv4 or valid IPv6 address. The BIG-IP system currently finds only exact matches for IP addresses. To use a route domain ID for a non-default route domain, that is, a route domain other than 0, append the character % and the route domain ID number to the end of the IP address. For example, to use route domain 2 with an IPv4 address of 1.1.1.1, you would type `1.1.1.1%2`. For the default route domain (0), do not append any additional characters to the IP address.

Viewing the IPsec diagnostics

Before you begin this task, you must create at least one IPsec tunnel through which you then transmit traffic.

You can view diagnostic statistics for any IPsec tunnel on the BIG-IP® system. This task describes searching by the traffic selector name, but you could also search by source and/or destination IP address or IPsec policy name.

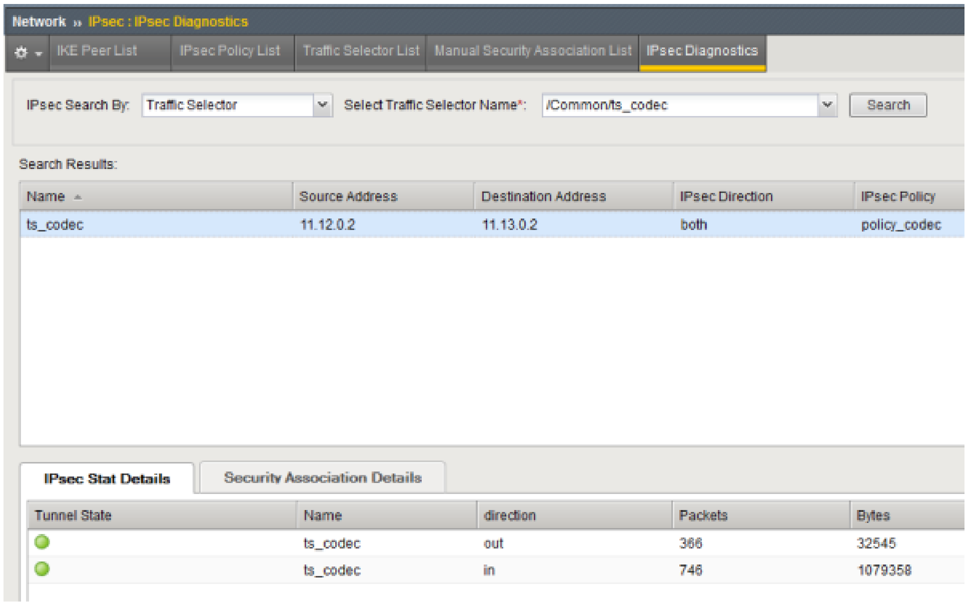
1. On the Main tab, click **Network** > **IPsec** > **IPsec Diagnostics**.
2. From the **IPsec Search By** list, select **Traffic Selector**.
   The search field label changes to **Select Traffic Selector Name**.
3. From the **Select Traffic Selector Name** list, select the name of the traffic selector that is associated with the communication channel you want to view, and click **Search**.
   The search results display the traffic selector you chose, including its source and destination addresses, direction, and associated IPsec policy.
4. Click the traffic selector.
   Additional details appear for that communication channel.
   - The IPsec Stat Details tab includes the tunnel state, direction, number of packets, and total bytes.
   - The Security Association Details tab includes the state of the association, source and destination IP addresses, direction, IPsec protocol, authentication algorithm, encryption algorithm, and SPI.

IPsec Diagnostics Example

These examples show the diagnostic details that are available as the result of an IPsec traffic selector search.

The color of the icon in the Tunnel State or security association (SA) State column indicates the condition of the connection.

- Green indicates that the tunnel is up and running.
- Blue indicates that the SA is in the negotiating phase, before the tunnel is up.
- Yellow indicates that the SA is still valid, but will be deleted soon.
- Red indicates that the tunnel is down.


Network » IPsec : IPsec Diagnostics
IKE Peer List
IPsec Policy List
Traffic Selector List
Manual Security Association List
IPsec Diagnostics
Search
IPsec Stat Details
Security Association Details

IPsec Search By: Traffic Selector

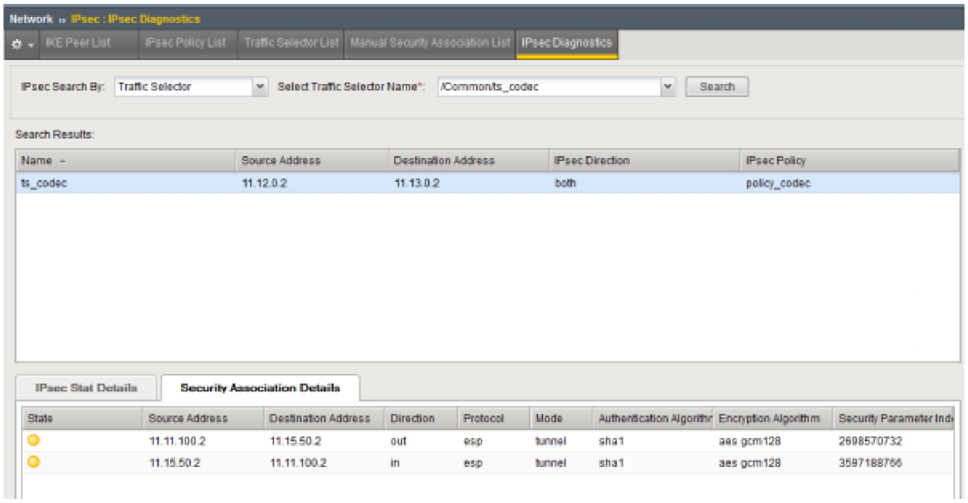
Select Traffic Selector Name*: /Common/ts_codec

Search Results:

Name	Source Address	Destination Address	IPsec Direction	IPsec Policy
ts_codec	11.12.0.2	11.13.0.2	both	policy_codec

Tunnel State	Name	direction	Packets	Bytes
●	ts_codec	out	366	32545
●	ts_codec	in	746	1079358

Figure 18: Example of IPsec Stat Details tab diagnostics


Network » IPsec : IPsec Diagnostics
IKE Peer List
IPsec Policy List
Traffic Selector List
Manual Security Association List
IPsec Diagnostics
Search
IPsec Stat Details
Security Association Details

IPsec Search By: Traffic Selector

Select Traffic Selector Name*: /Common/ts_codec

Search Results:

Name	Source Address	Destination Address	IPsec Direction	IPsec Policy
ts_codec	11.12.0.2	11.13.0.2	both	policy_codec

State	Source Address	Destination Address	Direction	Protocol	Mode	Authentication Algorithr	Encryption Algorithm	Security Parameter Ind
●	11.11.100.2	11.15.50.2	out	esp	tunnel	sha1	aes gcm128	2698570732
●	11.15.50.2	11.11.100.2	in	esp	tunnel	sha1	aes gcm128	3597188766

Figure 19: Example of IPsec Security Association Details tab diagnostics

Index