DEPLOYMENT GUIDE

Ruckus ICX FastIron IP Security Deployment Guide

Supporting FastIron 08.0.80

Part Number: 53-1005585-01
Publication Date: 28 November 2018

Copyright, Trademark and Proprietary Rights Information

Export Restrictions

These products and associated technical data (in print or electronic form) may be subject to export control laws of the United States of America. It is your responsibility to determine the applicable regulations and to comply with them. The following notice is applicable for all products or technology subject to export control:

These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. government or as otherwise authorized by U.S. law and regulations.

Disclaimer

THIS CONTENT AND ASSOCIATED PRODUCTS OR SERVICES ("MATERIALS"), ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT PERMISSIBLE PURSUANT TO APPLICABLE LAW, ARRIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, FREEDOM FROM COMPUTER VIRUS, AND WARRANTIES ARISING FROM COURSE OF DEALING OR COURSE OF PERFORMANCE. ARRIS does not represent or warrant that the functions described or contained in the Materials will be uninterrupted or error-free, that defects will be corrected, or are free of viruses or other harmful components. ARRIS does not make any warranties or representations regarding the use of the Materials in terms of their completeness, correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise. As a condition of your use of the Materials, you warrant to ARRIS that you will not make use thereof for any purpose that is unlawful or prohibited by their associated terms of use.

Limitation of Liability

IN NO EVENT SHALL ARRIS, ARRIS AFFILIATES, OR THEIR OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, SUPPLIERS, LICENSORS AND THIRD PARTY PARTNERS, BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER, EVEN IF ARRIS HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT, OR ANY OTHER THEORY ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIALS. Because some jurisdictions do not allow limitations on how long an implied warranty lasts, or the exclusion or limitation of liability for consequential or incidental damages, some of the above limitations may not apply to you.

Trademarks

ARRIS, the ARRIS logo, Ruckus, Ruckus Wireless, Ruckus Networks, Ruckus logo, the Big Dog design, BeamFlex, ChannelFly, EdgeIron, FastIron, HyperEdge, ICX, IronPoint, OPENG, SmartCell, Unleashed, Xclaim, ZoneFlex are trademarks of ARRIS International plc and/or its affiliates. Wi-Fi Alliance, Wi-Fi, the Wi-Fi logo, the Wi-Fi CERTIFIED logo, Wi-Fi Protected Access (WPA), the Wi-Fi Protected Setup logo, and WMM are registered trademarks of Wi-Fi Alliance. Wi-Fi Protected Setup™, Wi-Fi Multimedia™, and WPA2™ are trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

Contents

Preface

Purpose of This Document

This document provides technical details, configuration and deployment scenarios, and use cases for the FastIron IPsec solution.

Audience

This document will be useful for network designers, systems engineers, administrators, and customers.

Objectives

The objective of this guide is to assist the administrator in IPsec deployment and required configurations.

This deployment guide covers the following topics in depth:

- IPsec overview
- Acronyms
- Supported Algorithms
- Establishment of IPsec tunnel
- IPsec deployments and configurations
- Troubleshooting
- Scalability considerations
- Downgrade considerations

Document History

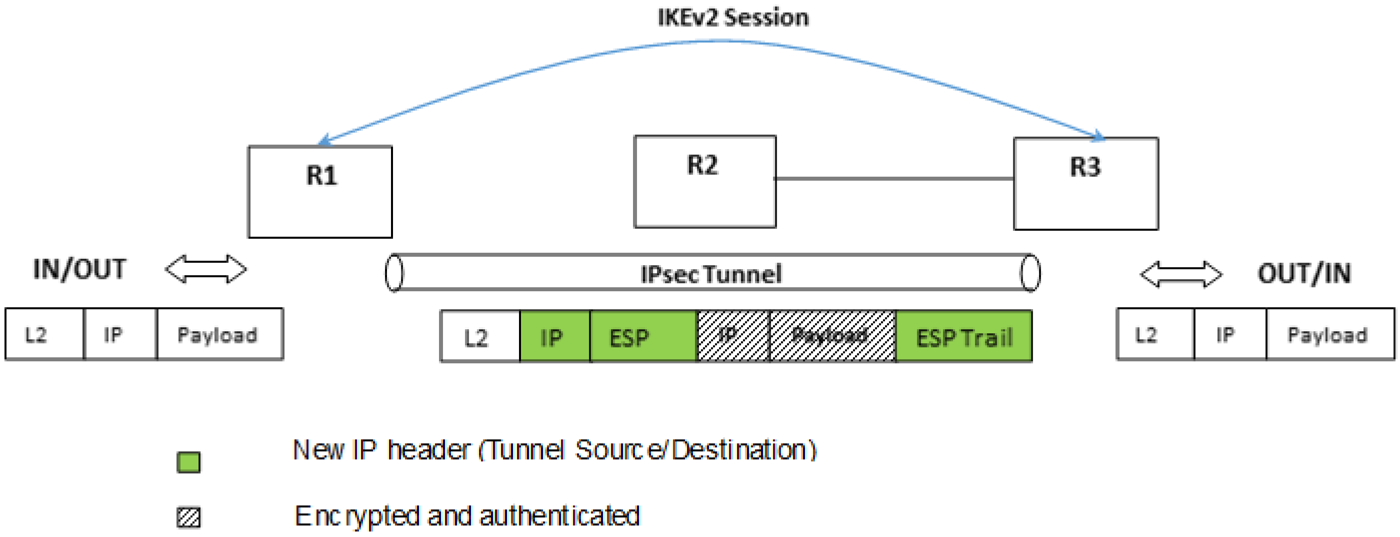
Date	Part Number	Description
November 28, 2018	53-1005585-01	Initial release.

IPsec Overview

IPsec introduces confidentiality and authentication services in the Ruckus routers. The IP crypto facilities supported today in FastIron are limited to authentication of control packets for OSPFv3 only. IPsec extends the crypto support to the data traffic and introduces data confidentiality along with the authentication services.

There are numerous applications for supporting IPsec for data traffic. The data confidentiality in federal deployments, VPN, and confidentiality in healthcare and campus are a few of the applications. The following figure explains the basic IPsec functionality.

FIGURE 1 Basic IPsec Functionality



Routers R1 and R3 want to securely transfer data over the public (unsecure) network. The administrator at R1 and R3 configures the IPsec tunnel parameters, the transform set, and the set of crypto algorithms to be used for encryption and authentication along with the filters for which these policies must be applied. Any packets matching this policy will be dropped until the tunnel is up. The tunnel establishment starts with the Internet Key Exchange (IKE) negotiations. In phase 1 of the IKE, a shared secret key is derived using Diffie-Hellman (DH) for encryption and decryption of the IKE packets. Subsequently, the IKE phase 2 derives the keys for data path encryption and decryption. The derived data path keys are programmed in the specialized hardware crypto engine, and the tunnel state is set to up. Once the tunnel is up, all the subsequent packets that match the filter or policy or the routes learned over the tunnel as the next hop will be encrypted and sent over the tunnel. A similar process is followed in the reverse direction to achieve the bidirectional data flow.

NOTE
The Ruckus ICX 7450 supports the following combinations:

- IPv4 payload over IPv4 tunnel
- IPv6 payload over IPv6 tunnel
- IPv6 payload over IPv4 tunnel

NOTE
IPv4 traffic can be routed over an IPv4 IPsec tunnel using RIP, OSPF, BGP, static route, and PBR.
IPv6 traffic can be routed over an IPv6 IPsec tunnel using RIPng, OSPFv3, BGP+, static route, and PBRv6.

NOTE
IPsec is only supported on the Ruckus ICX 7450 platform. To enable IPsec functionality, an ICX7400-SERVICE-MOD module must be installed on the device or stack. For further information about the installation procedure, refer to the *Ruckus ICX 7450 Switch Hardware Installation Guide*.
ICX7400-SERVICE-MOD modules are not supported in Ruckus ICX 7450 devices used as port extender (PE) units in a campus fabric setup.
Only one active ICX7400-SERVICE-MOD module is supported in a Ruckus ICX 7450 stack.

Acronyms

The following acronyms may be used in this document in the description of IP security (IPsec).

TABLE 1 IPsec-related Acronyms

Acronym	Full Form of Abbreviation
AES-128-GCM	Advanced Encryption Standard with 128-bit key size in Galois Counter Mode
AES-256-GCM	Advanced Encryption Standard with 256-bit key size in Galois Counter Mode
AES-128-CBC	Advanced Encryption Standard with 128-bit key size in Cipher Block Chaining Mode
AES-256-CBC	Advanced Encryption Standard with 256-bit key size in Cipher Block Chaining Mode
CA	Certificate Authority
CDP	CRL Distribution Point
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DH	Diffie-Hellman
DPD	Dead Peer Detection
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload protocol
IKEv2	Internet Key Exchange protocol, version 2
IPsec	IP Security
NAT	Network Address Translation
OSCP	Online Certificate Status Protocol
PAT	Port Address Translation
PKI	Public Key Infrastructure
PRF	Pseudo Random Function
RSA	Rivest-Shamir-Adleman
SA	Security Association
SAD	Security Association Database
SCEP	Simple Certificate Enrollment Protocol
SHA	Secure Hash Algorithm
SPD	Security Policy Database
SPI	Security Parameters Index (used to identify an SA)
TOS	Type of Service field in IPv4 header
VRF	Virtual Routing and Forwarding
VTI	Virtual Tunnel Interface

Supported Algorithms

TABLE 2 Supported Algorithms

Supported Algorithms	Encryption	Integrity	PRF	DH Group
IKEv2 Algorithms	AES-256-CBC	SHA-384	SHA-384	19
	AES-128-CBC	SHA-256	SHA-256	20
				14
Data Path Algorithms (Suite B Cryptographic)	AES-256-GCM			
	AES-128-GCM			

Establishment of IPsec Tunnel

IKEv2 negotiations are used to establish an IPsec tunnel. There are two phases in the IKEv2 negotiation process.

In phase 1, the tunnel endpoints exchange proposals for mutual authentication and securing the communication channel. The IKEv2 protocol is used to dynamically negotiate and authenticate keying material and other security parameters that are required to establish secure communications. A secret shared key for encrypting and decrypting the IKEv2 packets themselves is derived in this phase. When the phase 1 negotiations are successful, an IKEv2 SA, which contains the negotiated security encryption and keying material, is established. The IKEv2 SA is a secure "control channel" where keys and other information for protecting phase 2 IKEv2 negotiations are maintained. The IKEv2 SA established in phase 1 is bidirectional. Phase 2 communication is secure.

In phase 2, the tunnel endpoints exchange proposals for securing the data that is to be sent over the tunnel. When phase 2 negotiations are successful, a pair of IPsec SAs is established. An IPsec SA is unidirectional; one IPsec SA is needed for inbound traffic and one for outbound traffic. SAs that are negotiated by using the IKE SA (such as IPsec SAs) are also known as *child SAs*. The negotiated data path keys are then programmed into the specialized hardware crypto engine (ICX7400-SERVICE-MOD module), and the tunnel state is set to up. At this point, the user data can be exchanged through the encrypted tunnel.

The set of IPsec parameters describing an IPsec tunnel connection is known as an *IPsec security policy*. The IPsec security policy describes how both endpoints use the security services, such as encryption, hash algorithms, and Diffie-Hellman groups, for secure communication. Multiple security policies may be defined between IPsec security peers. For example, you can define an IPsec security policy that creates a tunnel between two hosts, and a different IPsec security policy that creates a tunnel between a host and a subnet, or between two subnets. Because multiple tunnels can exist between two peers, multiple IPsec SAs can be active at any time between two peers.

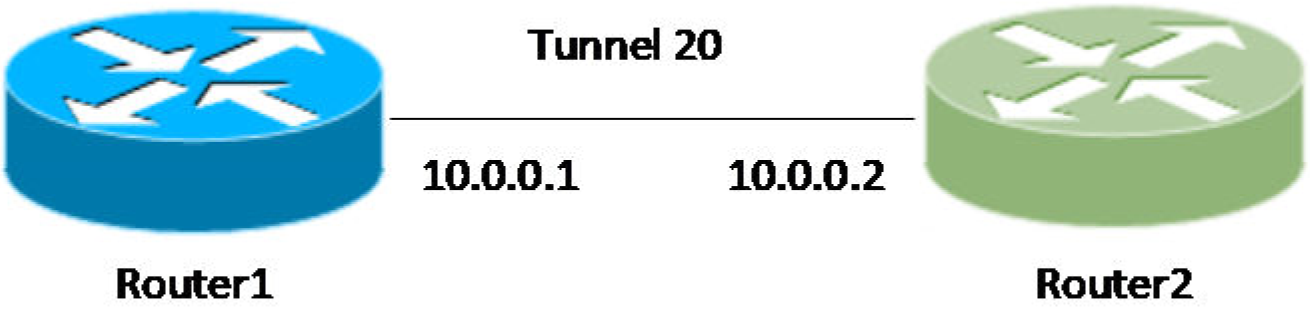
Deployments

Site-to-Site VPN: Configuration for an IPv4/IPv6 Over IPv4 IPsec Tunnel Using Default Settings

An IPsec tunnel is configured by binding an IPsec profile to the virtual tunnel interface (VTI) at each end of the IPsec tunnel. When the default settings for the IPsec profile are used, minimal configuration is needed to establish the tunnel.

In the following example, Router1 and Router2 are the devices at each end of the tunnel. On each device, a default IPsec profile (profA) is created and bound to the VTI by using the **tunnel protection ipsec profile** command.

FIGURE 2 Site-to-Site Deployment of IPsec



NOTE
Tunnel endpoints may be multiple hops away and the base path reachable over any interior gateway protocols such as static routing, RIP, OSPF, BGP, and so on.

NOTE
Source and destination addresses of the outer header of the tunneled packet can be:

- In a different VRF from the VRF for which the packet is received (including the default global VRF).
- In the same VRF that receives the packet.

IPsec Default Settings

By default, an IPsec profile has the following settings:

- Default IKEv2 profile: def-ike-profile
  - Default IKEv2 profile accepts any remote identity parameters
  - Uses default auth-proposal
- Default IPsec proposal: def-ipsec-prop
  - Protocol: ESP
  - Encryption Protocol: aes-gcm-256
- Default IKEv2 policy
  - VRF: match any
  - Local Address/Mask: 0.0.0.0/0.0.0.0 (match any)
  - Default IKEv2 Proposal: def-ike-prop
    - Encryption: aes256
    - Integrity: sha384
    - PRF: sha384
    - DH Group: 384_ECP/Group 20

Configurations

Router1

```
Router1(config)# ipsec profile profA
Router1(config-ipsec-profile-profA)# exit

Router1(config)# interface tunnel 20
Router1(config-tnif-20)# tunnel mode ipsec ipv4
Router1(config-tnif-20)# tunnel protection ipsec profile profA
Router1(config-tnif-20)# tunnel source 10.1.1.1
Router1(config-tnif-20)# tunnel destination 10.1.1.2
Router1(config-tnif-20)# ip address 10.0.0.1 255.255.255.0
Router1(config-tnif-20)# exit
```

Router2

```
Router2(config)# ipsec profile profA
Router2(config-ipsec-profile-profA)# exit

Router2(config)# interface tunnel 20
Router2(config-tnif-20)# tunnel mode ipsec ipv4
Router2(config-tnif-20)# tunnel protection ipsec profile profA
Router2(config-tnif-20)# tunnel source 10.1.1.2
Router2(config-tnif-20)# tunnel destination 10.1.1.1
Router2(config-tnif-20)# ip address 10.0.0.2 255.255.255.0
Router2(config-tnif-20)# exit
```

NOTE

IPv6 is supported over an IPv4 IPsec tunnel by configuring an IPv6 address under the tunnel interface.

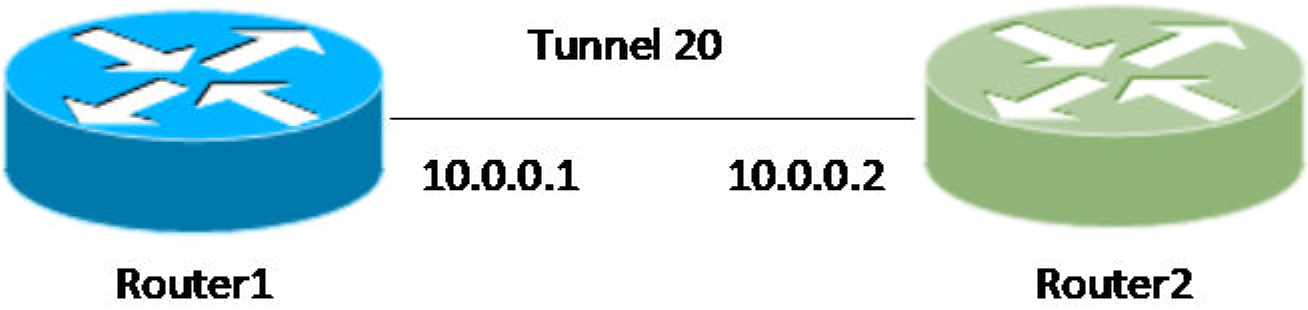
```
Router2(config)# interface tunnel 20
Router2(config-tnif-20)# tunnel mode ipsec ipv4
Router2(config-tnif-20)# tunnel protection ipsec profile profA
Router2(config-tnif-20)# tunnel source 10.1.1.2
Router2(config-tnif-20)# tunnel destination 10.1.1.1
Router2(config-tnif-20)# ip address 10.0.0.2 255.255.255.0
Router2(config-tnif-20)# ipv6 address 1000::1/64
Router2(config-tnif-20)# exit
```

Site-to-Site VPN: Configuration for an IPv4/IPv6 Over IPv4 IPsec Tunnel Using Nondefault Settings

An IPsec tunnel is configured by binding an IPsec profile to the virtual tunnel interface (VTI) at each end of the IPsec tunnel. With nondefault settings for the IPsec tunnel, the user-defined IKEv2 proposal, IKEv2 policy, IKEv2 auth-proposal, IKEv2 profile, IPsec proposal, and IPsec profile must be created at the two endpoints. The IKEv2 proposal is tied to the IKEv2 policy, the IKEv2 auth-proposal is tied to the IKEv2 profile, and the IKEv2 profile and IPsec proposal are tied to the IPsec profile.

In the following example, Router1 and Router2 are the devices at each end of the tunnel. On each device, a nondefault IPsec profile (profA) is created and bound to the VTI by using the **tunnel protection ipsec profile** command.

FIGURE 3 Site-to-Site Deployment of IPsec



NOTE
Tunnel endpoints may be multiple hops away and the base path reachable over any interior gateway protocols such as static routing, RIP, OSPF, BGP, and so on.

NOTE
Source and destination addresses of the outer header of the tunneled packet can be:

- In a different VRF from the VRF for which the packet is received (including the default global VRF).
- In the same VRF that receives the packet.

IPsec Nondefault Settings

The following nondefault settings are used to create an IPsec tunnel:

- IKEv2 auth-proposal: auth_propA
- IKEv2 profile: profA (All parameters are mandatory.)
  - local-identifier
  - remote-identifier
  - match-identity local: same as local-identifier
  - match-identity remote: same as remote-identifier
  - IKEv2 auth-proposal: auth_propA

NOTE
By default, an IKE profile can be used for tunnels in any VRF (including def-vrf). The IKEv2 profile can be protected and restricted for usage in a specific VRF using the **protected** *vrf-name* configuration.

- IKEv2 proposal: propA
  - Encryption: aes128
  - Integrity: sha256
  - PRF: sha256
  - DH Group: 256_ECP/Group 19
- IKEv2 policy: polA
  - VRF
  - Local Address/Mask
  - IKEv2 Proposal: propA
- IPsec proposal: propA
  - Protocol: ESP
  - Encryption: aes-gcm-128
- IPsec profile: profA
  - IPsec proposal: propA
  - IKEv2 profile: profA

Configurations

Router1

```
Router1(config)# ikev2 auth-proposal auth_propA
Router1(config-ike-auth-proposal-auth_propA)# pre-shared-key test123
Router1(config-ike-auth-proposal-auth_propA)# exit

Router1(config)# ikev2 proposal propA
Router1(config-ike-proposal-propA)# dhgroup 19
Router1(config-ike-proposal-propA)# no dhgroup 20
Router1(config-ike-proposal-propA)# prf sha256
Router1(config-ike-proposal-propA)# no prf sha384
Router1(config-ike-proposal-propA)# encryption aes-cbc-128
Router1(config-ike-proposal-propA)# no encryption aes-cbc-256
Router1(config-ike-proposal-propA)# integrity sha256
Router1(config-ike-proposal-propA)# no integrity sha384
Router1(config-ike-proposal-propA)# exit

Router1(config)# ikev2 profile profA
Router1(config-ike-profile-profA)# authentication auth_propA
Router1(config-ike-profile-profA)# local-identifier address 1.1.1.1
Router1(config-ike-profile-profA)# remote-identifier address 2.2.2.2
Router1(config-ike-profile-profA)# match-identity local address 1.1.1.1
Router1(config-ike-profile-profA)# match-identity remote address 2.2.2.2
Router1(config-ike-profile-profA)# exit

Router1(config)# ikev2 policy polA
Router1(config-ike-policy-polA)# proposal propA
Router1(config-ike-policy-polA)# match address-local 10.1.1.1 255.255.255.255
Router1(config-ike-policy-polA)# exit

Router1(config)# ipsec proposal propA
Router1(config-ipsec-proposal-propA)# encryption-algorithm aes-gcm-128
Router1(config-ipsec-proposal-propA)# no encryption-algorithm aes-gcm-256
Router1(config-ipsec-proposal-propA)# exit

Router1(config)# ipsec profile profA
Router1(config-ipsec-profile-profA)# proposal propA
Router1(config-ipsec-profile-profA)# ike-profile profA
Router1(config-ipsec-profile-profA)# exit

Router1(config)# interface tunnel 20
Router1(config-tnif-20)# tunnel mode ipsec ipv4
```

```
Router1(config-tnif-20)# tunnel protection ipsec profile profA
Router1(config-tnif-20)# tunnel source 10.1.1.1
Router1(config-tnif-20)# tunnel destination 10.1.1.2
Router1(config-tnif-20)# ip address 10.0.0.1 255.255.255.0
Router1(config-tnif-20)# exit
```

Router2

```
Router2(config)# ikev2 auth-proposal auth_propA
Router2(config-ike-auth-proposal-auth_propA)# pre-shared-key test123
Router2(config-ike-auth-proposal-auth_propA)# exit

Router2(config)# ikev2 proposal propA
Router2(config-ike-proposal-propA)# dhgroup 19
Router2(config-ike-proposal-propA)# no dhgroup 20
Router2(config-ike-proposal-propA)# prf sha256
Router2(config-ike-proposal-propA)# no prf sha384
Router2(config-ike-proposal-propA)# encryption aes-cbc-128
Router2(config-ike-proposal-propA)# no encryption aes-cbc-256
Router2(config-ike-proposal-propA)# integrity sha256
Router2(config-ike-proposal-propA)# no integrity sha384
Router2(config-ike-proposal-propA)# exit

Router2(config)# ikev2 profile profA
Router2(config-ike-profile-profA)# authentication auth_propA
Router2(config-ike-profile-profA)# local-identifier address 2.2.2.2
Router2(config-ike-profile-profA)# remote-identifier address 1.1.1.1
Router2(config-ike-profile-profA)# match-identity local address 2.2.2.2
Router2(config-ike-profile-profA)# match-identity remote address 1.1.1.1
Router2(config-ike-profile-profA)# exit

Router2(config)# ikev2 policy polA
Router2(config-ike-policy-polA)# proposal propA
Router2(config-ike-policy-polA)# match address-local 10.1.1.2 255.255.255.255
Router2(config-ike-policy-polA)# exit

Router2(config)# ipsec proposal propA
Router2(config-ipsec-proposal-propA)# encryption-algorithm aes-gcm-128
Router2(config-ipsec-proposal-propA)# no encryption-algorithm aes-gcm-256
Router2(config-ipsec-proposal-propA)# exit

Router2(config)# ipsec profile profA
Router2(config-ipsec-profile-profA)# proposal propA
Router2(config-ipsec-profile-profA)# ike-profile profA
Router2(config-ipsec-profile-profA)# exit

Router2(config)# interface tunnel 20
Router2(config-tnif-20)# tunnel mode ipsec ipv4
Router2(config-tnif-20)# tunnel protection ipsec profile profA
Router2(config-tnif-20)# tunnel source 10.1.1.2
Router2(config-tnif-20)# tunnel destination 10.1.1.1
Router2(config-tnif-20)# ip address 10.0.0.2 255.255.255.0
Router2(config-tnif-20)# exit
```

NOTE

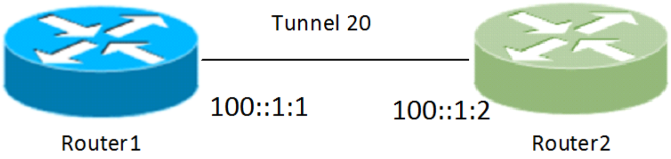
IPv6 is supported over an IPv4 IPsec tunnel by configuring an IPv6 address under the tunnel interface.

```
Router2(config)# interface tunnel 20
Router2(config-tnif-20)# tunnel mode ipsec ipv4
Router2(config-tnif-20)# tunnel protection ipsec profile profA
Router2(config-tnif-20)# tunnel source 10.1.1.2
Router2(config-tnif-20)# tunnel destination 10.1.1.1
Router2(config-tnif-20)# ip address 10.0.0.2 255.255.255.0
Router2(config-tnif-20)# ipv6 address 1000::1/64
Router2(config-tnif-20)# exit
```

Site-to-Site VPN: Configuration for an IPv6 Over IPv6 IPsec Tunnel Using Nondefault Settings

An IPsec tunnel is configured by binding an IPsec profile to the virtual tunnel interface (VTI) at each end of the IPsec tunnel. With nondefault settings for the IPsec tunnel, the user-defined IKEv2 proposal, IKEv2 policy, IKEv2 auth-proposal, IKEv2 profile, IPsec proposal, and IPsec profile must be created at the two endpoints. The IKEv2 proposal is tied to the IKEv2 policy, the IKEv2 auth-proposal is tied to the IKEv2 profile, and the IKEv2 profile and IPsec proposal are tied to the IPsec profile. In the following example, Router1 and Router2 are the devices at each end of the tunnel. On each device, a nondefault IPsec profile (profA) is created and bound to the VTI by using the **tunnel protection ipsec profile** command.

FIGURE 4 Site-to-Site Deployment of IPsec



NOTE
Tunnel endpoints may be multiple hops away and the base path reachable over any interior gateway protocols such as static routing, RIP, OSPF, BGP, and so on.

NOTE
Source and destination addresses of the outer header of the tunneled packet can be:

- In a different VRF from the VRF for which the packet is received (including the default global VRF).
- In the same VRF that receives the packet.

Configurations

Router1

```
Router1(config)# ikev2 auth-proposal auth_propA
Router1(config-ike-auth-proposal-auth_propA)# pre-shared-key test123
Router1(config-ike-auth-proposal-auth_propA)# exit

Router1(config)# ikev2 proposal propA
Router1(config-ike-proposal-propA)# dhgroup 19
Router1(config-ike-proposal-propA)# no dhgroup 20
Router1(config-ike-proposal-propA)# prf sha256
Router1(config-ike-proposal-propA)# no prf sha384
Router1(config-ike-proposal-propA)# encryption aes-cbc-128
Router1(config-ike-proposal-propA)# no encryption aes-cbc-256
Router1(config-ike-proposal-propA)# integrity sha256
Router1(config-ike-proposal-propA)# no integrity sha384
Router1(config-ike-proposal-propA)# exit

Router1(config)# ikev2 profile profA
Router1(config-ike-profile-profA)# authentication auth_propA
Router1(config-ike-profile-profA)# local-identifier address 1001::1:1
Router1(config-ike-profile-profA)# remote-identifier address 1002::1:1
Router1(config-ike-profile-profA)# match-identity local address 1001::1:1
Router1(config-ike-profile-profA)# match-identity remote address 1002::1:1
Router1(config-ike-profile-profA)# exit
```

```
Router1(config)# ikev2 policy polA
Router1(config-ike-policy-polA)# proposal propA
Router1(config-ike-policy-polA)# match address-local 100::1:1/64
Router1(config-ike-policy-polA)# exit

Router1(config)# ipsec proposal propA
Router1(config-ipsec-proposal-propA)# encryption-algorithm aes-gcm-128
Router1(config-ipsec-proposal-propA)# no encryption-algorithm aes-gcm-256
Router1(config-ipsec-proposal-propA)# exit

Router1(config)# ipsec profile profA
Router1(config-ipsec-profile-profA)# proposal propA
Router1(config-ipsec-profile-profA)# ike-profile profA
Router1(config-ipsec-profile-profA)# exit

Router1(config)# interface tunnel 20
Router1(config-tnif-20)# tunnel mode ipsec ipv6
Router1(config-tnif-20)# tunnel protection ipsec profile profA
Router1(config-tnif-20)# tunnel source 100::1:1
Router1(config-tnif-20)# tunnel destination 100::1:2
Router1(config-tnif-20)# ip address 1000::1/64
Router1(config-tnif-20)# exit
```

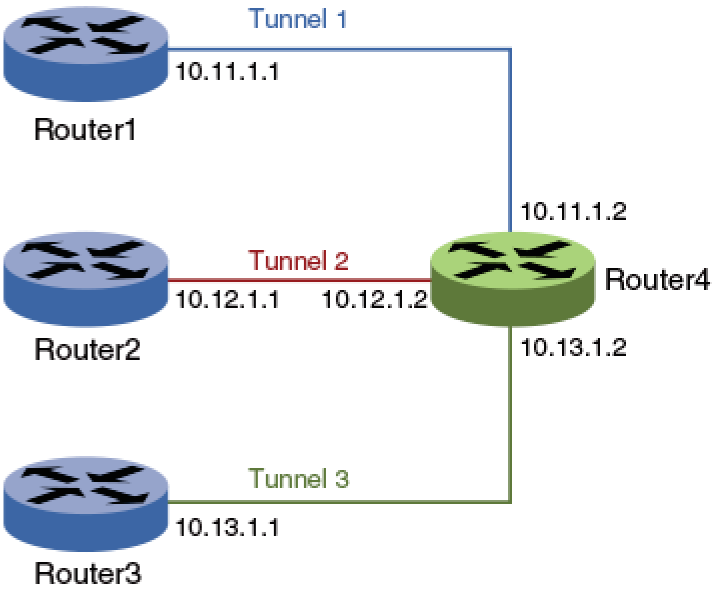
Router2

```
Router2(config)# ikev2 auth-proposal auth_propA
Router2(config-ike-auth-proposal-auth_propA)# pre-shared-key test123
Router2(config-ike-auth-proposal-auth_propA)# exit

Router2(config)# ikev2 proposal propA
Router2(config-ike-proposal-propA)# dhgroup 19
Router2(config-ike-proposal-propA)# no dhgroup 20
Router2(config-ike-proposal-propA)# prf sha256
Router2(config-ike-proposal-propA)# no prf sha384
Router2(config-ike-proposal-propA)# encryption aes-cbc-128
Router2(config-ike-proposal-propA)# no encryption aes-cbc-256
Router2(config-ike-proposal-propA)# integrity sha256
Router2(config-ike-proposal-propA)# no integrity sha384
Router2(config-ike-proposal-propA)# exit

Router2(config)# ikev2 profile profA
Router2(config-ike-profile-profA)# authentication auth_propA
Router2(config-ike-profile-profA)# local-identifier address 1002::1:1
Router2(config-ike-profile-profA)# remote-identifier address 1001::1:1
Router2(config-ike-profile-profA)# match-identity local address 1002::1:1
Router2(config-ike-profile-profA)# match-identity remote address 1001::1:1
Router2(config-ike-profile-profA)# exit

Router2(config)# ikev2 policy polA
Router2(config-ike-policy-polA)# proposal propA
Router2(config-ike-policy-polA)# match address-local 10.1.1.2 255.255.255.255
Router2(config-ike-policy-polA)# exit

Router2(config)# ipsec proposal propA
Router2(config-ipsec-proposal-propA)# encryption-algorithm aes-gcm-128
Router2(config-ipsec-proposal-propA)# no encryption-algorithm aes-gcm-256
Router2(config-ipsec-proposal-propA)# exit

Router2(config)# ipsec profile profA
Router2(config-ipsec-profile-profA)# proposal propA
Router2(config-ipsec-profile-profA)# ike-profile profA
Router2(config-ipsec-profile-profA)# exit

Router2(config)# interface tunnel 20
Router2(config-tnif-20)# tunnel mode ipsec ipv4
Router2(config-tnif-20)# tunnel protection ipsec profile profA
Router2(config-tnif-20)# tunnel source 10.1.1.2
Router2(config-tnif-20)# tunnel destination 10.1.1.1
Router2(config-tnif-20)# ip address 10.0.0.2 255.255.255.0
Router2(config-tnif-20)# exit
```

IPsec Hub-to-Spoke VPN

In an IPsec hub-to-spoke deployment, spokes communicate with each other by way of the hub router (there is no direct communication). One IPsec tunnel is stitched from each spoke to the hub. During spoke-to-spoke communication, the hub receives traffic from one spoke over one IPsec tunnel and forwards the traffic to another spoke over another IPsec tunnel. As shown in the following figure, when Router1 and Router2 (spokes) want to communicate, Router4 (hub) receives traffic from Router1 over IPsec Tunnel 1 and forwards it to Router2 over IPsec Tunnel 2. From the hub (Router4) perspective, this is also referred to as *tunnel stitching*.

FIGURE 5 Hub-to-Spoke Deployment of IPsec



NOTE
Tunnel endpoints may be multiple hops away and the base path reachable over any interior gateway protocols such as static routing, RIP, OSPF, BGP, and so on.

NOTE
The Ruckus ICX 7450 has a 50 percent performance degradation when used in a tunnel stitching deployment.

Configurations

In the following configuration example, the IPsec tunnels are running in the user VRF (vrf1) and the base path is in the default VRF. In this case, the IGP running in the basepath is OSPF.

Router1

```
Router1(config)# router ospf
Router1(config-router-ospf-vrf-default-vrf)# area 0
Router1(config-router-ospf-vrf-default-vrf)# exit


Router1(config)# interface loopback 1
Router1(config-lbif-1)# ip address 10.100.100.1 255.255.255.255
Router1(config-lbif-1)# ip ospf area 0
Router1(config-lbif-1)# exit
```

```
Router1(config)# ikev2 proposal ikev2_propA
Router1(config-ike-proposal-ikev2_propA)# exit

Router1(config)# ikev2 auth-proposal ikev2_auth_propA
Router1(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router1(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router1(config)# ikev2 policy ikev2_policyA
Router1(config-ike-policy-ikev2_policyA)# proposal ikev2_proposal
Router1(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.1 255.255.255.255
Router1(config-ike-policy-ikev2_policyA)# exit

Router1(config)# ikev2 profile ikev2_profA
Router1(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router1(config-ike-profile-ikev2_profA)# local-identifier address 10.1.1.1
Router1(config-ike-profile-ikev2_profA)# remote-identifier address 10.4.4.4
Router1(config-ike-profile-ikev2_profA)# match-identity local address 10.1.1.1
Router1(config-ike-profile-ikev2_profA)# match-identity remote address 10.4.4.4
Router1(config-ike-profile-ikev2_profA)# exit

Router1(config)# ipsec proposal ipsec_propA
Router1(config-ipsec-proposal-ipsec_propA)# exit

Router1(config)# ipsec profile ipsec_profA
Router1(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router1(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router1(config-ipsec-profile-ipsec_profA)# exit

Router1(config)# interface tunnel 1
Router1(config-tnif-1)# vrf forwarding vrf1
Router1(config-tnif-1)# tunnel mode ipsec ipv4
Router1(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router1(config-tnif-1)# tunnel source loopback 1
Router1(config-tnif-1)# tunnel destination 10.100.100.4
Router1(config-tnif-1)# ip address 10.11.1.1 255.255.255.252
Router1(config-tnif-1)# exit
```

Router2

```
Router2(config)# router ospf
Router2(config-router-ospf-vrf-default-vrf)# area 0
Router2(config-router-ospf-vrf-default-vrf)# end

Router2(config)# interface loopback 1
Router2(config-lbif-1)# ip address 10.100.100.2 255.255.255.255
Router2(config-lbif-1)# ip ospf area 0
Router2(config-lbif-1)# exit

Router2(config)# ikev2 proposal ikev2_propA
Router2(config-ike-proposal-ikev2_propA)# exit

Router2(config)# ikev2 auth-proposal ikev2_auth_propA
Router2(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router2(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router2(config)# ikev2 policy ikev2_policyA
Router2(config-ike-policy-ikev2_policyA)# proposal ikev2_proposal
Router2(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.2 255.255.255.255
Router2(config-ike-policy-ikev2_policyA)# exit

Router2(config)# ikev2 profile ikev2_profA
Router2(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router2(config-ike-profile-ikev2_profA)# local-identifier address 10.2.2.2
Router2(config-ike-profile-ikev2_profA)# remote-identifier address 10.4.4.4
Router2(config-ike-profile-ikev2_profA)# match-identity local address 10.2.2.2
Router2(config-ike-profile-ikev2_profA)# match-identity remote address 10.4.4.4
Router2(config-ike-profile-ikev2_profA)# exit

Router2(config)# ipsec proposal ipsec_propA
Router2(config-ipsec-proposal-ipsec_propA)# exit
```

```
Router2(config)# ipsec profile ipsec_profA
Router2(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router2(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router2(config-ipsec-profile-ipsec_profA)# exit

Router2(config)# interface tunnel 1
Router2(config-tnif-1)# vrf forwarding vrf1
Router2(config-tnif-1)# tunnel mode ipsec ipv4
Router2(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router2(config-tnif-1)# tunnel source loopback 1
Router2(config-tnif-1)# tunnel destination 10.100.100.4
Router2(config-tnif-1)# ip address 10.12.1.1 255.255.255.252
Router2(config-tnif-1)# exit
```

Router3

```
Router3(config)# router ospf
Router3(config-router-ospf-vrf-default-vrf)# area 0
Router3(config-router-ospf-vrf-default-vrf)# end

Router3(config)# interface loopback 1
Router3(config-lbif-1)# ip address 10.100.100.3 255.255.255.255
Router3(config-lbif-1)# ip ospf area 0
Router3(config-lbif-1)# exit

Router3(config)# ikev2 proposal ikev2_propA
Router3(config-ike-proposal-ikev2_propA)# exit

Router3(config)# ikev2 auth-proposal ikev2_auth_propA
Router3(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router3(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router3(config)# ikev2 policy ikev2_policyA
Router3(config-ike-policy-ikev2_policyA)# proposal ikev2_proposal
Router3(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.3 255.255.255.255
Router3(config-ike-policy-ikev2_policyA)# exit

Router3(config)# ikev2 profile ikev2_profA
Router3(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router3(config-ike-profile-ikev2_profA)# local-identifier address 10.3.3.3
Router3(config-ike-profile-ikev2_profA)# remote-identifier address 10.4.4.4
Router3(config-ike-profile-ikev2_profA)# match-identity local address 10.3.3.3
Router3(config-ike-profile-ikev2_profA)# match-identity remote address 10.4.4.4
Router3(config-ike-profile-ikev2_profA)# exit

Router3(config)# ipsec proposal ipsec_propA
Router3(config-ipsec-proposal-ipsec_propA)# exit

Router3(config)# ipsec profile ipsec_profA
Router3(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router3(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router3(config-ipsec-profile-ipsec_profA)# exit

Router3(config)# interface tunnel 1
Router3(config-tnif-1)# vrf forwarding vrf1
Router3(config-tnif-1)# tunnel mode ipsec ipv4
Router3(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router3(config-tnif-1)# tunnel source loopback 1
Router3(config-tnif-1)# tunnel destination 10.100.100.4
Router3(config-tnif-1)# ip address 10.13.1.1 255.255.255.252
Router3(config-tnif-1)# exit
```

Router4

NOTE
Router4, which is performing IPsec tunnel stitching, can be any device that supports IPsec. The following example shows how to configure Router4 when the device is a Ruckus ICX 7450 switch.

```
Router4(config)# router ospf
Router4(config-router-ospf-vrf-default-vrf)# area 0
Router4(config-router-ospf-vrf-default-vrf)# end

Router4(config)# interface loopback 1
Router4(config-lbif-1)# ip address 10.100.100.4 255.255.255.255
Router4(config-lbif-1)# exit

Router4(config)# ikev2 proposal ikev2_propA
Router4(config-ike-proposal-ikev2_propA)# exit

Router4(config)# ikev2 auth-proposal ikev2_auth_propB
Router4(config-ike-auth-proposal-ikev2_auth_propB)# pre-shared-key ps_key
Router4(config-ike-auth-proposal-ikev2_auth_propB)# exit

Router4(config)# ikev2 auth-proposal ikev2_auth_propC
Router4(config-ike-auth-proposal-ikev2_auth_propC)# pre-shared-key ps_key
Router4(config-ike-auth-proposal-ikev2_auth_propC)# exit

Router4(config)# ikev2 auth-proposal ikev2_auth_propD
Router4(config-ike-auth-proposal-ikev2_auth_propD)# pre-shared-key ps_key
Router4(config-ike-auth-proposal-ikev2_auth_propD)# exit

Router4(config)# ikev2 policy ikev2_policyA
Router4(config-ike-policy-ikev2_policyA)# proposal ikev2_propA
Router4(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.4 255.255.255.255
Router4(config-ike-policy-ikev2_policyA)# exit

Router4(config)# ikev2 profile ikev2_profB
Router4(config-ike-profile-ikev2_profB)# authentication ikev2_auth_propB
Router4(config-ike-profile-ikev2_profB)# local-identifier address 10.4.4.4
Router4(config-ike-profile-ikev2_profB)# remote-identifier address 10.1.1.1
Router4(config-ike-profile-ikev2_profB)# match-identity local address 10.4.4.4
Router4(config-ike-profile-ikev2_profB)# match-identity remote address 10.1.1.1
Router4(config-ike-profile-ikev2_profB)# exit

Router4(config)# ikev2 profile ikev2_profC
Router4(config-ike-profile-ikev2_profC)# authentication ikev2_auth_propC
Router4(config-ike-profile-ikev2_profC)# local-identifier address 10.4.4.4
Router4(config-ike-profile-ikev2_profC)# remote-identifier address 10.2.2.2
Router4(config-ike-profile-ikev2_profC)# match-identity local address 10.4.4.4
Router4(config-ike-profile-ikev2_profC)# match-identity remote address 10.2.2.2
Router4(config-ike-profile-ikev2_profC)# exit

Router4(config)# ikev2 profile ikev2_profD
Router4(config-ike-profile-ikev2_profD)# authentication ikev2_auth_propD
Router4(config-ike-profile-ikev2_profD)# local-identifier address 10.4.4.4
Router4(config-ike-profile-ikev2_profD)# remote-identifier address 10.3.3.3
Router4(config-ike-profile-ikev2_profD)# match-identity local address 10.4.4.4
Router4(config-ike-profile-ikev2_profD)# match-identity remote address 10.3.3.3
Router4(config-ike-profile-ikev2_profD)# exit

Router4(config)# ipsec proposal ipsec_propA
Router4(config-ipsec-proposal-ipsec_propA)# exit

Router4(config)# ipsec profile ipsec_profB
Router4(config-ipsec-profile-ipsec_profB)# proposal ipsec_propA
Router4(config-ipsec-profile-ipsec_profB)# ike-profile ikev2_profB
Router4(config-ipsec-profile-ipsec_profB)# exit

Router4(config)# ipsec profile ipsec_profC
Router4(config-ipsec-profile-ipsec_profC)# proposal ipsec_propA
Router4(config-ipsec-profile-ipsec_profC)# ike-profile ikev2_profC
Router4(config-ipsec-profile-ipsec_profC)# exit
```

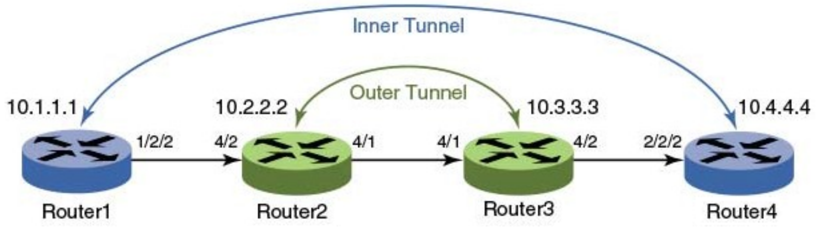
```
Router4(config)# ipsec profile ipsec_profD
Router4(config-ipsec-profile-ipsec_profD)# proposal ipsec_propA
Router4(config-ipsec-profile-ipsec_profD)# ike-profile ikev2_profD
Router4(config-ipsec-profile-ipsec_profD)# exit

Router4(config)# interface tunnel 1
Router4(config-tnif-1)# vrf forwarding vrf1
Router4(config-tnif-1)# tunnel mode ipsec ipv4
Router4(config-tnif-1)# tunnel protection ipsec profile ipsec_profB
Router4(config-tnif-1)# tunnel source loopback 1
Router4(config-tnif-1)# tunnel destination 10.100.100.1
Router4(config-tnif-1)# ip address 10.11.1.2 255.255.255.252
Router4(config-tnif-1)# exit

Router4(config)# interface tunnel 2
Router4(config-tnif-2)# vrf forwarding vrf1
Router4(config-tnif-2)# tunnel mode ipsec ipv4
Router4(config-tnif-2)# tunnel protection ipsec profile ipsec_profC
Router4(config-tnif-2)# tunnel source loopback 1
Router4(config-tnif-2)# tunnel destination 10.100.100.2
Router4(config-tnif-2)# ip address 10.12.1.2 255.255.255.252
Router4(config-tnif-2)# exit

Router4(config)# interface tunnel 3
Router4(config-tnif-3)# vrf forwarding vrf1
Router4(config-tnif-3)# tunnel mode ipsec ipv4
Router4(config-tnif-3)# tunnel protection ipsec profile ipsec_profD
Router4(config-tnif-3)# tunnel source loopback 1
Router4(config-tnif-3)# tunnel destination 10.100.100.3
Router4(config-tnif-3)# ip address 10.13.1.2 255.255.255.252
Router4(config-tnif-3)# exit
```

IPv4/IPv6 Over IPv4 IPsec Tunnel in IPsec Tunnel

IPsec tunnel in IPsec tunnel is a double-encryption use case. In the base path of one IPsec tunnel (inner tunnel) is another IPsec tunnel (outer tunnel). The tunnel endpoints of the inner tunnel are reachable by way of another IPsec tunnel (outer tunnel). While traffic traverses between outer IPsec tunnel endpoints, the traffic, already encrypted once by the inner IPsec tunnel, is encrypted a second time by the outer IPsec tunnel. Thus, there are two levels of encryption.

In the following figure, the inner IPsec tunnel starts at Router1 and ends at Router4; the outer IPsec tunnel starts at Router2 and ends at Router3. At Router1, plain host traffic is routed over the inner IPsec tunnel and forwarded as encrypted towards Router2 after the first level of encryption. At Router2, because the inner tunnel endpoint (Router4) is reachable by way of the outer tunnel, traffic goes through the second level of encryption and is forwarded out towards Router3. At Router3, traffic is decrypted and forwarded towards Router4 with the first level of encryption completed at Router1. At Router4, traffic is completely decrypted and forwarded as plain host traffic towards its destination.

FIGURE 6 Tunnel in Tunnel Deployment of IPsec



NOTE
Tunnel endpoints may be multiple hops away and the base path reachable over any interior gateway protocols such as static routing, RIP, OSPF, BGP, and so on.

Configurations

In the following configuration example, the inner tunnel is running in the user VRF (vrf1) and the outer tunnel is running in the default VRF.

Router1

```
Router1(config)# router ospf
Router1(config-router-ospf-vrf-default-vrf)# area 0
Router1(config-router-ospf-vrf-default-vrf)# end

Router1(config)# interface loopback 1
Router1(config-lbif-1)# ip address 10.100.100.1 255.255.255.255
Router1(config-lbif-1)# ip ospf area 0
Router1(config-lbif-1)# exit

Router1(config)# ikev2 proposal ikev2_propA
Router1(config-ike-proposal-ikev2_propA)# exit

Router1(config)# ikev2 auth-proposal ikev2_auth_propA
Router1(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router1(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router1(config)# ikev2 policy ikev2_policyA
Router1(config-ike-policy-ikev2_policyA)# proposal ikev2_propA
Router1(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.1 255.255.255.255
Router1(config-ike-policy-ikev2_policyA)# exit

Router1(config)# ikev2 profile ikev2_profA
Router1(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router1(config-ike-profile-ikev2_profA)# local-identifier address 10.1.1.1
Router1(config-ike-profile-ikev2_profA)# remote-identifier address 10.4.4.4
Router1(config-ike-profile-ikev2_profA)# match-identity local address 10.1.1.1
Router1(config-ike-profile-ikev2_profA)# match-identity remote address 10.4.4.4
Router1(config-ike-profile-ikev2_profA)# exit

Router1(config)# ipsec proposal ipsec_propA
Router1(config-ipsec-proposal-ipsec_propA)# exit

Router1(config)# ipsec profile ipsec_profA
Router1(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router1(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router1(config-ipsec-profile-ipsec_profA)# exit

Router1(config)# interface tunnel 1
Router1(config-tnif-1)# vrf forwarding vrf1
Router1(config-tnif-1)# tunnel mode ipsec ipv4
Router1(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router1(config-tnif-1)# tunnel source loopback 1
Router1(config-tnif-1)# tunnel destination 10.100.100.4
Router1(config-tnif-1)# ip address 10.11.1.1 255.255.255.252
Router1(config-tnif-1)# exit
```

Router2

```
Router2(config)# router ospf
Router2(config-router-ospf-vrf-default-vrf)# area 0
Router2(config-router-ospf-vrf-default-vrf)# end

Router2(config)# interface loopback 1
Router2(config-lbif-1)# ip address 10.100.100.2 255.255.255.255
Router2(config-lbif-1)# exit
```

```
Router2(config)# ikev2 proposal ikev2_propA
Router2(config-ike-proposal-ikev2_propA)# exit

Router2(config)# ikev2 auth-proposal ikev2_auth_propA
Router2(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router2(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router2(config)# ikev2 policy ikev2_policyA
Router2(config-ike-policy-ikev2_policyA)# proposal ikev2_propA
Router2(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.2 255.255.255.255
Router2(config-ike-policy-ikev2_policyA)# exit

Router2(config)# ikev2 profile ikev2_profA
Router2(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router2(config-ike-profile-ikev2_profA)# local-identifier address 10.2.2.2
Router2(config-ike-profile-ikev2_profA)# remote-identifier address 10.3.3.3
Router2(config-ike-profile-ikev2_profA)# match-identity local address 10.2.2.2
Router2(config-ike-profile-ikev2_profA)# match-identity remote address 10.3.3.3
Router2(config-ike-profile-ikev2_profA)# exit

Router2(config)# ipsec proposal ipsec_propA
Router2(config-ipsec-proposal-ipsec_propA)# exit

Router2(config)# ipsec profile ipsec_profA
Router2(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router2(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router2(config-ipsec-profile-ipsec_profA)# exit

Router2(config)# interface tunnel 1
Router2(config-tnif-1)# tunnel mode ipsec ipv4
Router2(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router2(config-tnif-1)# tunnel source loopback 1
Router2(config-tnif-1)# tunnel destination 10.100.100.3
Router2(config-tnif-1)# ip address 10.12.1.1 255.255.255.252
Router2(config-tnif-1)# ip ospf area 0
Router2(config-tnif-1)# exit
```

Router3

```
Router3(config)# router ospf
Router3(config-router-ospf-vrf-default-vrf)# area 0
Router3(config-router-ospf-vrf-default-vrf)# end

Router3(config)# interface loopback 1
Router3(config-lbif-1)# ip address 10.100.100.3 255.255.255.255
Router3(config-lbif-1)# exit

Router3(config)# ikev2 proposal ikev2_propA
Router3(config-ike-proposal-ikev2_propA)# exit

Router3(config)# ikev2 auth-proposal ikev2_auth_propA
Router3(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router3(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router3(config)# ikev2 policy ikev2_policyA
Router3(config-ike-policy-ikev2_policyA)# proposal ikev2_propA
Router3(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.3 255.255.255.255
Router3(config-ike-policy-ikev2_policyA)# exit

Router3(config)# ikev2 profile ikev2_profA
Router3(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router3(config-ike-profile-ikev2_profA)# local-identifier address 10.3.3.3
Router3(config-ike-profile-ikev2_profA)# remote-identifier address 10.2.2.2
Router3(config-ike-profile-ikev2_profA)# match-identity local address 10.3.3.3
Router3(config-ike-profile-ikev2_profA)# match-identity remote address 10.2.2.2
Router3(config-ike-profile-ikev2_profA)# exit

Router3(config)# ipsec proposal ipsec_propA
Router3(config-ipsec-proposal-ipsec_propA)# exit

Router3(config)# ipsec profile ipsec_profA
```

```
Router3(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router3(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router3(config-ipsec-profile-ipsec_profA)# exit

Router3(config)# interface tunnel 1
Router3(config-tnif-1)# tunnel mode ipsec ipv4
Router3(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router3(config-tnif-1)# tunnel source loopback 1
Router3(config-tnif-1)# tunnel destination 10.100.100.2
Router3(config-tnif-1)# ip address 10.12.1.2 255.255.255.252
Router3(config-tnif-1)# ip ospf area 0
Router3(config-tnif-1)# exit
```

Router4

```
Router4(config)# router ospf
Router4(config-router-ospf-vrf-default-vrf)# area 0
Router4(config-router-ospf-vrf-default-vrf)# end


Router4(config)# interface loopback 1
Router4(config-lbif-1)# ip address 10.100.100.4 255.255.255.255
Router4(config-lbif-1)# ip ospf area 0
Router4(config-lbif-1)# exit

Router4# configure terminal
Router4(config)# ikev2 proposal ikev2_propA
Router4(config-ike-proposal-ikev2_propA)# exit

Router4(config)# ikev2 auth-proposal ikev2_auth_propA
Router4(config-ike-auth-proposal-ikev2_auth_propA)# pre-shared-key ps_key
Router4(config-ike-auth-proposal-ikev2_auth_propA)# exit

Router4(config)# ikev2 policy ikev2_policyA
Router4(config-ike-policy-ikev2_policyA)# proposal ikev2_propA
Router4(config-ike-policy-ikev2_policyA)# match address-local 10.100.100.4 255.255.255.255
Router4(config-ike-policy-ikev2_policyA)# exit

Router4(config)# ikev2 profile ikev2_profA
Router4(config-ike-profile-ikev2_profA)# authentication ikev2_auth_propA
Router4(config-ike-profile-ikev2_profA)# local-identifier address 10.4.4.4
Router4(config-ike-profile-ikev2_profA)# remote-identifier address 10.1.1.1
Router4(config-ike-profile-ikev2_profA)# match-identity local address 10.4.4.4
Router4(config-ike-profile-ikev2_profA)# match-identity remote address 10.1.1.1
Router4(config-ike-profile-ikev2_profA)# exit

Router4(config)# ipsec proposal ipsec_propA
Router4(config-ipsec-proposal-ipsec_propA)# exit

Router4(config)# ipsec profile ipsec_profA
Router4(config-ipsec-profile-ipsec_profA)# proposal ipsec_propA
Router4(config-ipsec-profile-ipsec_profA)# ike-profile ikev2_profA
Router4(config-ipsec-profile-ipsec_profA)# exit

Router4(config)# interface tunnel 1
Router4(config-tnif-1)# vrf forwarding vrf1
Router4(config-tnif-1)# tunnel mode ipsec ipv4
Router4(config-tnif-1)# tunnel protection ipsec profile ipsec_profA
Router4(config-tnif-1)# tunnel source loopback 1
Router4(config-tnif-1)# tunnel destination 10.100.100.1
Router4(config-tnif-1)# ip address 10.11.1.2 255.255.255.252
Router4(config-tnif-1)# exit
```

PKI Support for IPsec

The Public Key Infrastructure (PKI) provides a security infrastructure for entities to ensure secured communication.

Each PKI peer holds a Digital Certificate which holds multiple attributes that ensure the entity can be trusted and can support secured communication.

PKI uses an asymmetric encryption algorithm; two different keys are used to encrypt and decrypt data. The key pair consists of a private key and a public key. The private key must be kept secret while the public key can be distributed. Data encrypted by one of the two keys can only be decrypted by the other. Data encrypted with a public key cannot be decrypted using a public key and vice versa. Users of a public key can be confident that the associated private key is owned by the correct remote subject.

FIGURE 7 Public Key Infrastructure

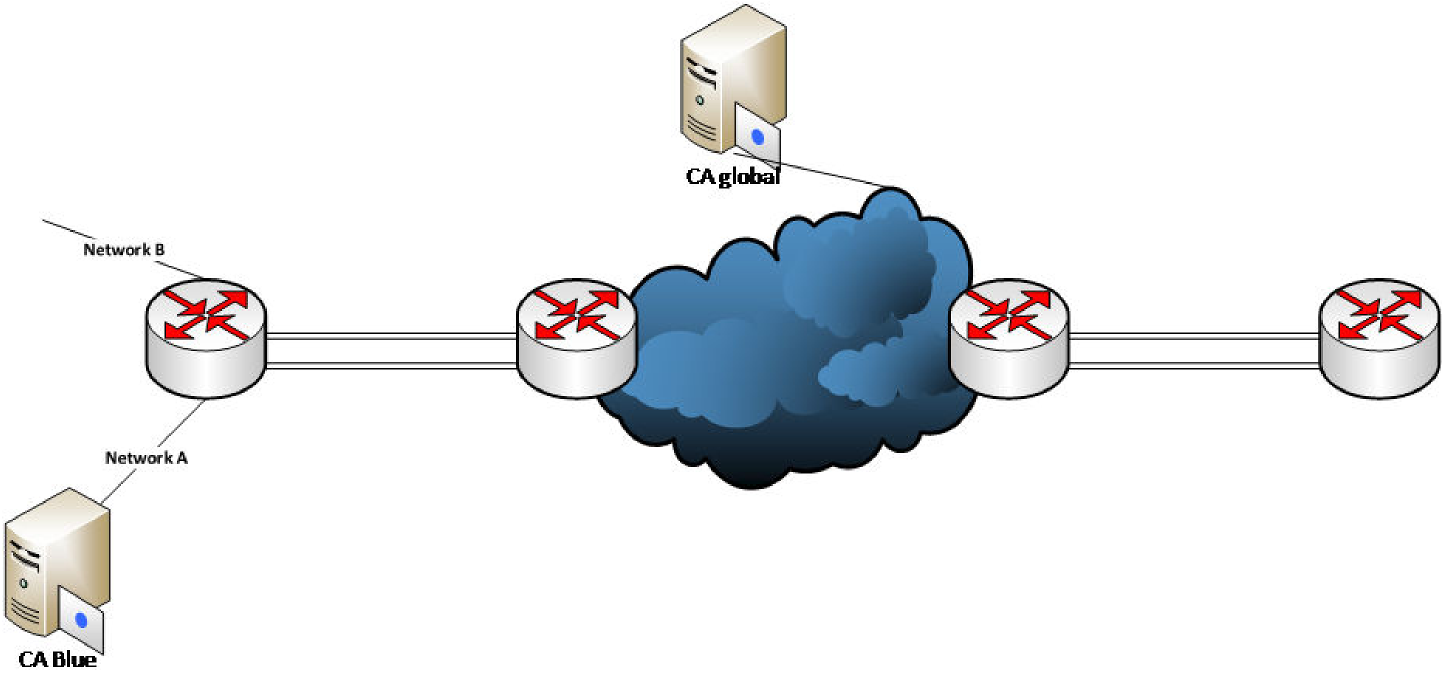
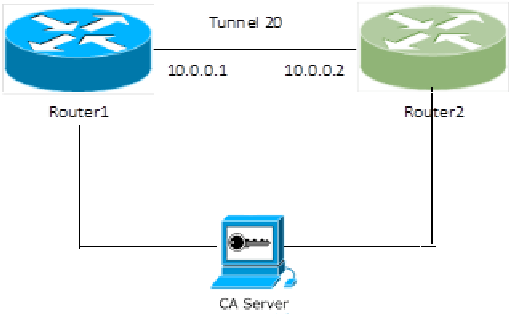


FIGURE 8 Site-to-Site VPN Using PKI



For authentication of the peers, digitally signed certificates can be used. You can create PKI entities for use in certificate authentication. To participate in certificate authentication with a Certificate Authority (CA), PKI entities must be enrolled. Entities can be enrolled through an automatic enrollment process, which allows them to send a certificate signing request (CSR) and to receive the required X.509 certificates from the CA in response. Entities can also be enrolled manually.

Configuring automatic enrollment of an entity (using SCEP) to the CA server involves the following tasks:

- Generate a cryptographic key using either the EC (elliptical key pair) or the RSA key pair option for PKI.
- Create a PKI entity.
- Configure the PKI profile.
- Configure the PKI trustpoint.
- Authenticate the PKI trustpoint.
- Enroll the entity to the PKI trustpoint.

Configuring PKI manual import of digital certificates involves the following components:

- Root CA certificate, which will be used with the configured trustpoint on the TOE
- The intermediate CA certificate
- The local DUT certificate, which is signed by the Root CA or any intermediate CA of the Root CA
- The key for the DUT local certificate

The listed certificates and keys must be copied into flash on the TOE.

Revocation Check for Peer Certificates Using CRL

A Certificate Revocation List (CRL) is a list of certificates signed by the CA that are prematurely invalid.

A periodic CRL timer runs, and each time it expires, it dumps the entire list of revocation information. The revocation check is performed when the CRL information is downloaded for the first time. When the subsequent timer expires, the revocation check is not performed unless the tunnels are forced to renegotiate.

The **revocation-check crl** command is used to set CRL as the revocation type.

```
device(config-pki-trustpoint-trust1)# revocation-check crl
```

The **show pki crl** command displays the downloaded revocation information.

```
device# show pki crl trustpoint_name
```

The **clear pki crl** command is used to clear the downloaded revocation information.

```
device(config)# clear pki crl trustpoint_name
```

The **pki export crl** command is used to export the CRL file of a given trustpoint. The following example exports the CRL for the trustpoint trust1 to the file crl_file.

```
device(config)# pki export crl trust1 url crl_file
```

CRL Distribution Point

The CRL Distribution Point (CDP) is used to retrieve the latest CRL of a CA. The CDP is usually located on an LDAP server or HTTP (web) server and is normally expressed as an ldap://host/dir or http://host/path URL.

After the revocation list is validated by the CRL server and the certificate is valid, then the IPsec session will come up between peers.

If the CRL server is not reachable and the response is not received during the IPsec tunnel bringup time, then the IPsec session does not come up between the peers.

NOTE

If for some reason the CRL server is not reachable, the IPsec peers will not check the revocation list until any of the IPsec peers reload or the IPsec session is cleared.

Configurations for PKI Trustpoint Using Revocation Check Using CRL

```
device# configure terminal
device(config)# pki trustpoint abcd
device(config-pki-trustpoint-abcd)# revocation-check crl
device(config-pki-trustpoint-abcd)# crl-query http://10.177.148.229/CertEnroll/FISQA-WIN12-01-CA-4.crl
device(config-pki-trustpoint-abcd)# fingerprint 3C:EA:EC:E6:F1:DD:3B:86:65:DE:58:F4:A2:75:D8:63:6D:23:68:40
device(config-pki-trustpoint-abcd)# exit
```

Revocation Check for Peer Certificates

FastIron devices support OCSP for checking the revocation status of a certificate received from a peer. The revocation status is checked for both the intermediate and the peer certificate. The following actions are taken in revocation checks:

- The OCSP signing bit should be set for the OCSP responder.
- If the OCSP responder responds that the certificate status is good, the IPsec session comes up.
- If the OCSP responder responds that the certificate status is revoked, the IPsec session does not come up.
- If the OCSP server is not reachable and the response is not received, then the IPsec session does not come up between the peers.

Configurations for Manual Enrollment

```
device# configure terminal
device(config)# crypto key generate rsa
device(config)# pki import key rsa rsakey pem url flash: dut.key.pem
device(config)# pki import abcd pem url flash: rootca.pem
device(config)# pki import abcd pem url flash: localcert.pem

device(config)# pki trustpoint abcd
device(config-pki-trustpoint-abcd)# ocsp http post
device(config-pki-trustpoint-abcd)# revocation-check ocsp
device(config-pki-trustpoint-abcd)# ocsp-url http://10.21.40.39:2560
device(config-pki-trustpoint-abcd)# fingerprint 3C:EA:EC:E6:F1:DD:3B:86:65:DE:58:F4:A2:75:D8:63:6D:23:68:40
device(config-pki-trustpoint-abcd)# exit

device(config)# pki authenticate abcd
device(config)# pki cert-validate abcd
PKI: Successfully validated the local certificate for trustpoint: abcd
```

Configurations for Automatic Enrollment

```
device(config)# pki entity entity1
device(config-pki-entity-entity1)# common-name "tester1"
device(config-pki-entity-entity1)# country-name "IN"
device(config-pki-entity-entity1)# state-name "KA"
device(config-pki-entity-entity1)# org-unit-name "FI"
device(config-pki-entity-entity1)# org-name "BRCD"

device(config)# pki profile-enrollment profile1
device(config-pki-profile-enrollment-profile1)# authentication-url http://WINN6C3R0LUDAJ.
englab.ruckus.com/CertSrv/mscep/mscep.dll
device(config-pki-profile-enrollment-profile1)# authentication-command WINN6C3R0LUDAJ.
```

```
englab.ruckus.com_englab-WIN-N6C3R0LUDAJ-CA-15
device(config-pki-profile-enrollment-profile1)# enrollment-url http://WINN6C3R0LUDAJ.
englab.ruckus.com/CertSrv/mscep/mscep.dll
device(config-pki-profile-enrollment-profile1)# password DB6E1F091AEF0244
device(config-pki-profile-enrollment-profile1)# exit

device(config)# pki trustpoint trust1
device(config-pki-trustpoint-trust1)# auto-enroll
device(config-pki-trustpoint-trust1)# enrollment retry-period 2
device(config-pki-trustpoint-trust1)# enrollment profile profile1
device(config-pki-trustpoint-trust1)# pki-entity entity1
device(config-pki-trustpoint-trust1)# eckeypair key-label eckeyAuto
device(config-pki-trustpoint-trust1)# fingerprint 36:0c:92:6e:df:b2:72:eb:59:e8:63:73:2a:98:a8:91:cb:
50:94:d9
device(config-pki-trustpoint-trust1)# revocation-check ocsp
device(config-pki-trustpoint-trust1)# ocsp-url http://10.21.40.39:2560
device(config-pki-trustpoint-trust1)# fingerprint 3C:EA:EC:E6:F1:DD:3B:
device(config)# pki authenticate trust1
device(config)# pki enroll trust1
```

IKE Configuration with PKI

```
device(config)# ikev2 auth-proposal withCert
device(config-ike-auth-proposal-withCert)# method remote rsa
device(config-ike-auth-proposal-withCert)# method local rsa
device(config-ike-auth-proposal-withCert)# pki-trustpoint abcd-CA sign
device(config-ike-auth-proposal-withCert)# pki-trustpoint abcd-CA verify
device(config-ike-auth-proposal-withCert)# exit
!
device(config)# ikev2 profile with_standalone
device(config-ike-profile-with_standalone)# authentication withCert
device(config-ike-profile-with_standalone)# local-identifier dn "CN=SPATHA_STANDALONE, ST=CA, C=US,
O=SPATHA_SPATHA_ORG_NAME, OU=SPATHA_ALONE_ORG_UNIT"
device(config-ike-profile-with_standalone)# remote-identifier dn "CN=SPATHA48F, ST=CA, C=US,
O=SPATHA48F_ORG_NAME"
device(config-ike-profile-with_standalone)# match-identity local dn "CN=SPATHA_STANDALONE, ST=CA, C=US,
O=SPATHA_SPATHA_ORG_NAME, OU=SPATHA_ALONE_ORG_UNIT"
device(config-ike-profile-with_standalone)# match-identity remote dn "CN=SPATHA48F, ST=CA, C=US,
O=SPATHA48F_ORG_NAME"
device(config-ike-profile-with_standalone)# exit
!
device(config)# ipsec profile withKey-L2
device(config-ipsec-profile-withKey-L2)# ike-profile withKey-L2
device(config-ipsec-profile-withKey-L2)# exit
!
device(config)# ipsec profile withCert
device(config-ipsec-profile-withCert)# ike-profile with_standalone
```

IPsec Tunnel Configuration

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode ipsec ipv4
device(config-tnif-1)# tunnel protection ipsec profile withCert
device(config-tnif-1)# tunnel source 10.1.1.1
device(config-tnif-1)# tunnel destination 10.1.1.2
device(config-tnif-1)# ip address 10.0.0.1 255.255.255.0
```

Show Commands

```
device# show pki certificates local
----------------PKI LOCAL CERTIFICATE ENTRY-----------------
CA: TLS-ABCD
Certificate:
Data:
Version: 3 (0x2)
```

```
Serial Number: 4125 (0x101d)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com
Validity
Not Before: Nov 7 02:24:18 2017 GMT
Not After : Nov 17 02:24:18 2018 GMT
Subject: CN=DUTFIPSCC, ST=CA, C=US/emailAddress=user@arris.com, O=DUTFIPSCC, OU=SQA
device#
device# show pki certificates trustpoint
----------------PKI TRUSTPOINT CERTIFICATE ENTRY-----------------
CA: TLS-ABCD
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
bd:fa:4f:da:bd:89:4a:5d
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com
Validity
Not Before: Nov 7 02:10:00 2017 GMT
Not After : Nov 7 02:10:00 2022 GMT
Subject: C=US, ST=CA, L=SJ, O=ROOTCA-CC, OU=SQA, CN=ROOTCA-CC/emailAddress=user@arris.com

device(config-tnif-1)# show ike certificate
Trustpoint: trustRSA
Local:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4100 (0x1004)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IN/ST=KA/L=Bangalore/O=Ruckus Arris/OU=NEBU/CN=ROOT RSA
        Validity
            Not Before: Feb 23 16:19:43 2018 GMT
            Not After : Feb 21 16:19:43 2028 GMT
        Subject: CN=ICX RSA/ST=KA/C=IN/O=NEBU/OU=Ruckus Arris
CA:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e2:11:82:3f:37:c2:6f:c0
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IN/ST=KA/L=Bangalore/O=Ruckus Arris/OU=NEBU/CN=ROOT RSA
        Validity
            Not Before: Feb 23 05:38:11 2018 GMT
            Not After : Feb 23 05:38:11 2023 GMT
        Subject: C=IN/ST=KA/L=Bangalore/O=Ruckus Arris/OU=NEBU/CN=ROOT RSA
Peer Certificates:
Cert Verified: True
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4101 (0x1005)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=Intermediate CA RSA/ST=KA/C=IN/O=Ruckus Arris/OU=NEBU
        Validity
            Not Before: Feb 23 16:22:11 2018 GMT
            Not After : Feb 21 16:22:11 2028 GMT
        Subject: CN=Linux RSA/ST=KA/C=IN/O=Ruckus Arris/OU=NEBU
Peer CAs:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4098 (0x1002)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IN/ST=KA/L=Bangalore/O=Ruckus Arris/OU=NEBU/CN=ROOT RSA
        Validity
            Not Before: Feb 23 12:13:26 2018 GMT
            Not After : Feb 21 12:13:26 2028 GMT
        Subject: CN=Intermediate CA RSA/ST=KA/C=IN/O=Ruckus Arris/OU=NEBU
device#
```

IPsec Over NAT

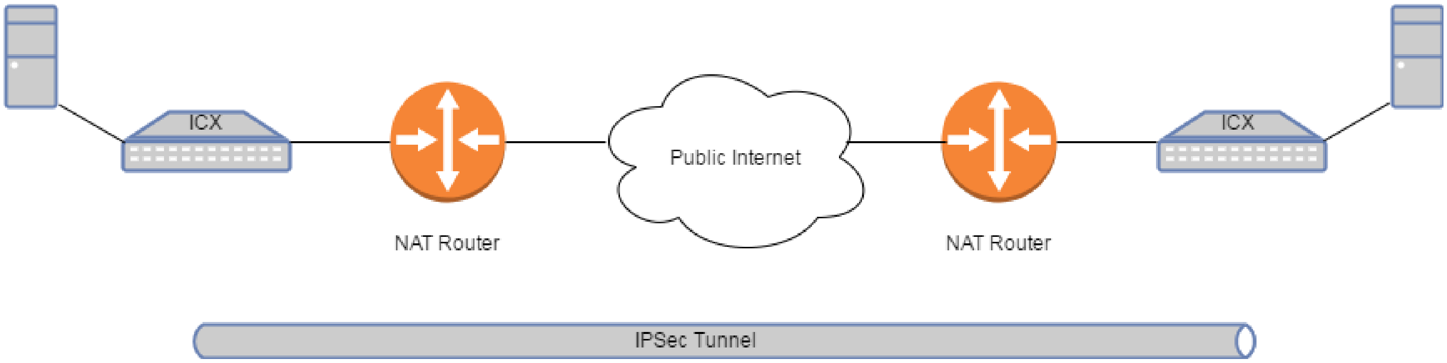
Network Address Translation (NAT) is a method to remap a private IP address to a public IP address by modifying the address information in the IP packet. This is done by the transit routers when the traffic passes through them.

This method allows you to limit the number of public IP addresses that could be owned by a user. In basic NAT (one-to-one NAT), there is one-to-one mapping of IP addresses. However, an effective NAT method is Port Address Translation (PAT), where many private addresses will map to a single public IP address.

An IPsec ESP packet does not contain port information like TCP and UDP. Therefore, a NAT (PAT) device is unable to do mapping and drops the packet. This is overcome by NAT Traversal (IPsec over NAT), which encapsulates the ESP packet inside a UDP header.

NAT Traversal is enabled by default. In FIPS mode, NAT Traversal cannot be disabled. In non-FIPS mode, you can disable NAT Traversal on an as-needed basis.

FIGURE 9 Deployment of ICX as IPsec Endpoints with NAT Traversal Enabled



NOTE
One of the primary use cases of IPsec is the remote access to a corporate network for telecommuters and employees accessing through a VPN. And this use case relies on NAT. Therefore, it is essential to provide NAT Traversal for IPsec. On the receiving end, the hashes of the IP address and port are calculated in the packet header. These are compared with the hashes received as part of the NAT-D payload. If they are different, it means the packet has undergone NAT. After this negotiation, the IPsec tunnel will be established with UDP encapsulation. In the UDP encapsulation, both source and destination ports will be set to 4500.
In UDP encapsulation, an ESP packet is treated like a UDP packet and NAT is applied normally without affecting the ESP packet inside.

NOTE
An SPI value of 0 (zero) must not be used in the SPI field of ESP packets. This is to distinguish IKE and ESP packets.

NAT Configuration

NAT is enabled by default. To disable NAT, use the **ikev2 nat-disable** command.

```
device(config)# ikev2 nat-disable
```

The **no** form of the command enables NAT.

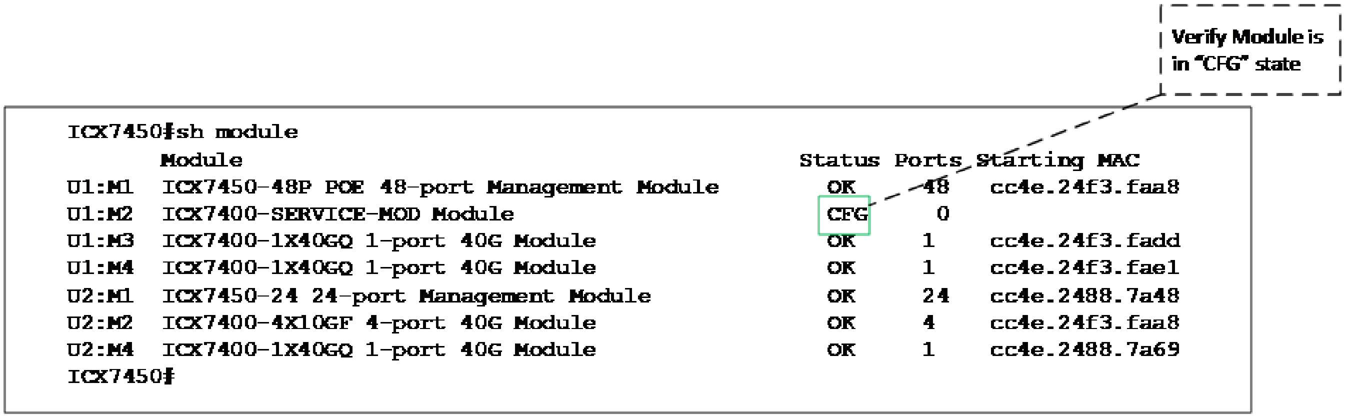
```
device(config)# no ikev2 nat-disable
```

IPsec Module Failure and Replacement

Complete the following steps to recover from an IPsec module failure.

1. Confirm the ICX7400-SERVICE-MOD module failure status (CFG) on the stack unit/device on which the module resides.

FIGURE 10 Confirming Module Failure

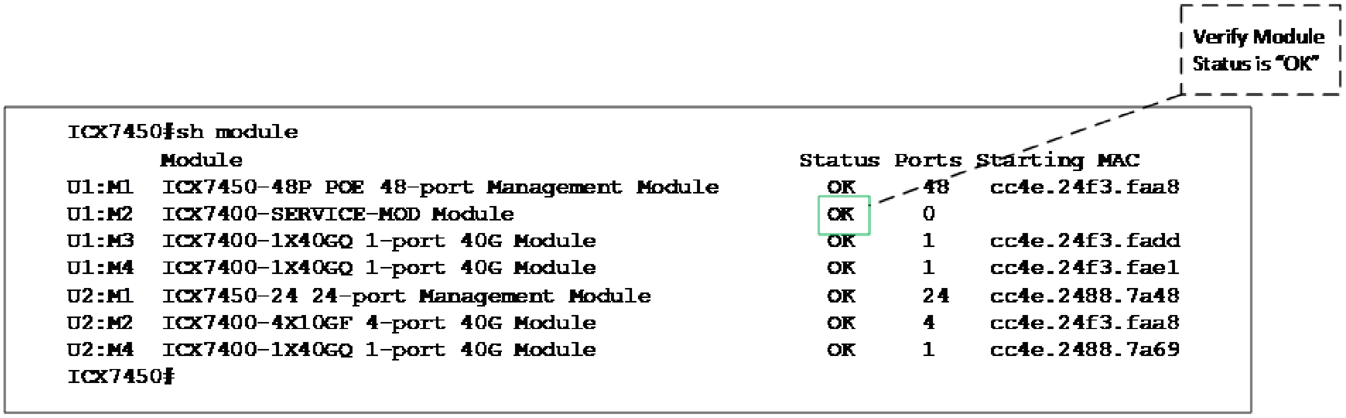


Verify Module is in "CFG" state

```
ICX7450#sh module
       Module                                        Status  Ports  Starting MAC
U1:M1  ICX7450-48P POE 48-port Management Module       OK     48    cc4e.24f3.faa8
U1:M2  ICX7400-SERVICE-MOD Module                      CFG     0
U1:M3  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.24f3.fadd
U1:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.24f3.fae1
U2:M1  ICX7450-24 24-port Management Module            OK     24    cc4e.2488.7a48
U2:M2  ICX7400-4X10GF 4-port 40G Module                OK      4    cc4e.24f3.faa8
U2:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.2488.7a69
ICX7450#
```

2. Power-off the stack unit/device of the failed ICX7400-SERVICE-MOD module.
3. Insert the new ICX7400-SERVICE-MOD module.
4. Power-on the stack unit/device and wait for it to come up.
5. Verify the new ICX7400-SERVICE-MOD module status (OK) on the stack unit/device where it is inserted.

FIGURE 11 Verifying Module Status



Verify Module Status is "OK"

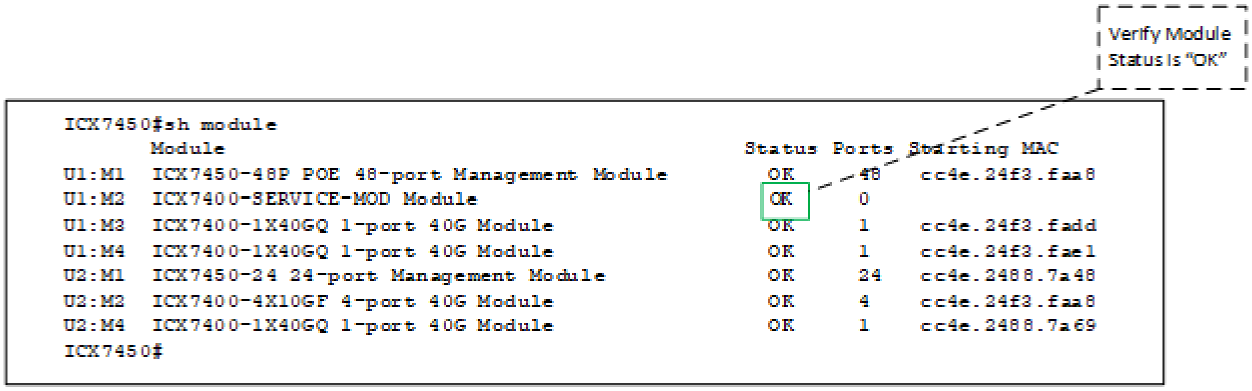
```
ICX7450#sh module
       Module                                        Status  Ports  Starting MAC
U1:M1  ICX7450-48P POE 48-port Management Module       OK     48    cc4e.24f3.faa8
U1:M2  ICX7400-SERVICE-MOD Module                      OK      0
U1:M3  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.24f3.fadd
U1:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.24f3.fae1
U2:M1  ICX7450-24 24-port Management Module            OK     24    cc4e.2488.7a48
U2:M2  ICX7400-4X10GF 4-port 40G Module                OK      4    cc4e.24f3.faa8
U2:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1    cc4e.2488.7a69
ICX7450#
```

Troubleshooting

Commonly Used Show Commands

The following show commands can be used in basic IPsec troubleshooting, which helps in checking IPsec tunnel health and triaging IPsec issues.

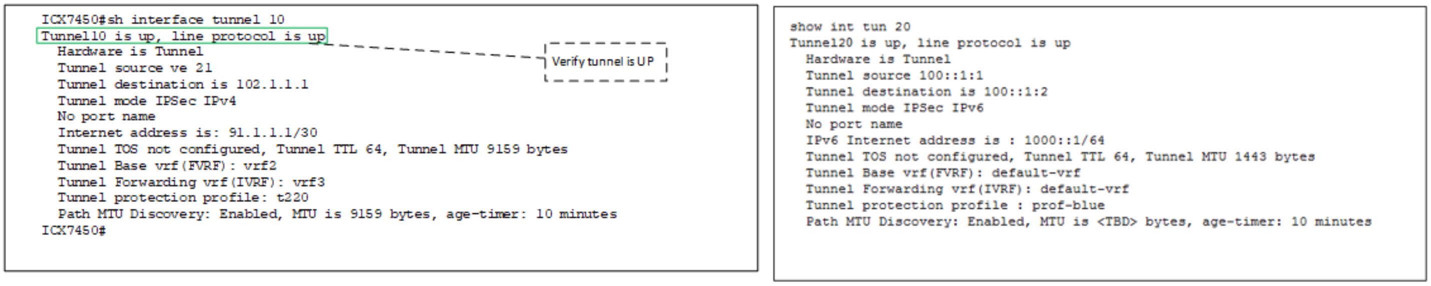
FIGURE 12 Show Module



```
ICX7450#sh module
       Module                                         Status  Ports  Starting MAC
U1:M1  ICX7450-48P POE 48-port Management Module       OK      48    cc4e.24f3.faa8
U1:M2  ICX7400-SERVICE-MOD Module                      OK      0
U1:M3  ICX7400-1X40GQ 1-port 40G Module                OK      1     cc4e.24f3.fadd
U1:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1     cc4e.24f3.fae1
U2:M1  ICX7450-24 24-port Management Module            OK      24    cc4e.2488.7a48
U2:M2  ICX7400-4X10GF 4-port 40G Module                OK      4     cc4e.24f3.faa8
U2:M4  ICX7400-1X40GQ 1-port 40G Module                OK      1     cc4e.2488.7a69
ICX7450#
```

Verify Module Status Is "OK"

FIGURE 13 Show Interface Tunnel for IPv4 and IPv6 Tunnels

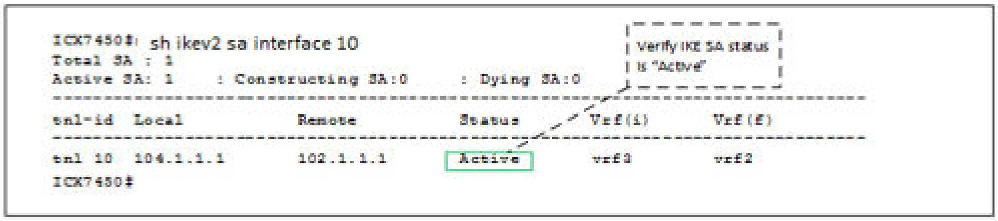


```
ICX7450#sh interface tunnel 10
Tunnel10 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source ve 21
  Tunnel destination is 102.1.1.1
  Tunnel mode IPSec IPv4
  No port name
  Internet address is: 91.1.1.1/30
  Tunnel TOS not configured, Tunnel TTL 64, Tunnel MTU 9159 bytes
  Tunnel Base vrf(FVRF): vrf2
  Tunnel Forwarding vrf(IVRF): vrf3
  Tunnel protection profile: t220
  Path MTU Discovery: Enabled, MTU is 9159 bytes, age-timer: 10 minutes
ICX7450#
```

Verify tunnel is UP

```
show int tun 20
Tunnel20 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source 100::1:1
  Tunnel destination is 100::1:2
  Tunnel mode IPSec IPv6
  No port name
  IPv6 Internet address is : 1000::1/64
  Tunnel TOS not configured, Tunnel TTL 64, Tunnel MTU 1443 bytes
  Tunnel Base vrf(FVRF): default-vrf
  Tunnel Forwarding vrf(IVRF): default-vrf
  Tunnel protection profile : prof-blue
  Path MTU Discovery: Enabled, MTU is <TBD> bytes, age-timer: 10 minutes
```

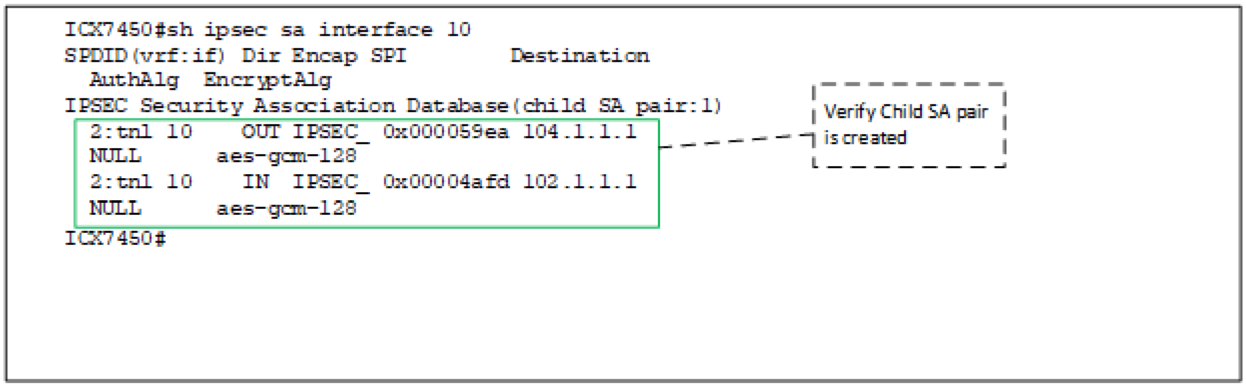
FIGURE 14 Show IKEv2 SA Interface



```
ICX7450#: sh ikev2 sa interface 10
Total SA : 1
Active SA: 1     : Constructing SA:0      : Dying SA:0
--------------------------------------------------------------------------------
tnl-id  Local            Remote          Status      Vrf(i)      Vrf(f)
--------------------------------------------------------------------------------
tnl 10  104.1.1.1        102.1.1.1       Active      vrf3        vrf2
ICX7450#
```

Verify IKE SA status is "Active"

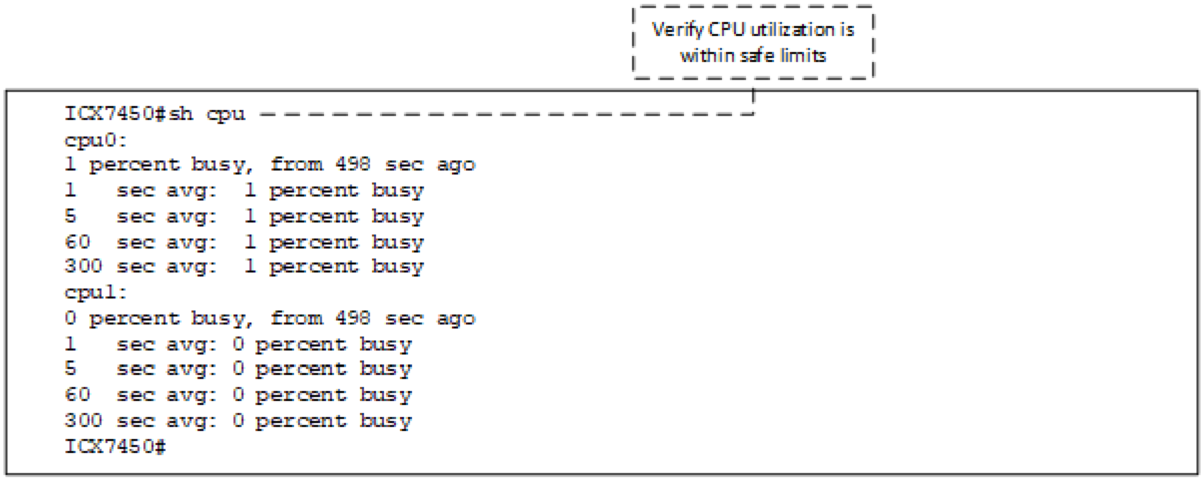
FIGURE 15 Show IPsec SA Interface



```
ICX7450#sh ipsec sa interface 10
SPDID(vrf:if) Dir Encap SPI        Destination
  AuthAlg  EncryptAlg
IPSEC Security Association Database(child SA pair:1)
  2:tnl 10    OUT IPSEC_ 0x000059ea 104.1.1.1
  NULL      aes-gcm-128
  2:tnl 10    IN  IPSEC_ 0x00004afd 102.1.1.1
  NULL      aes-gcm-128
ICX7450#
```

Verify Child SA pair is created

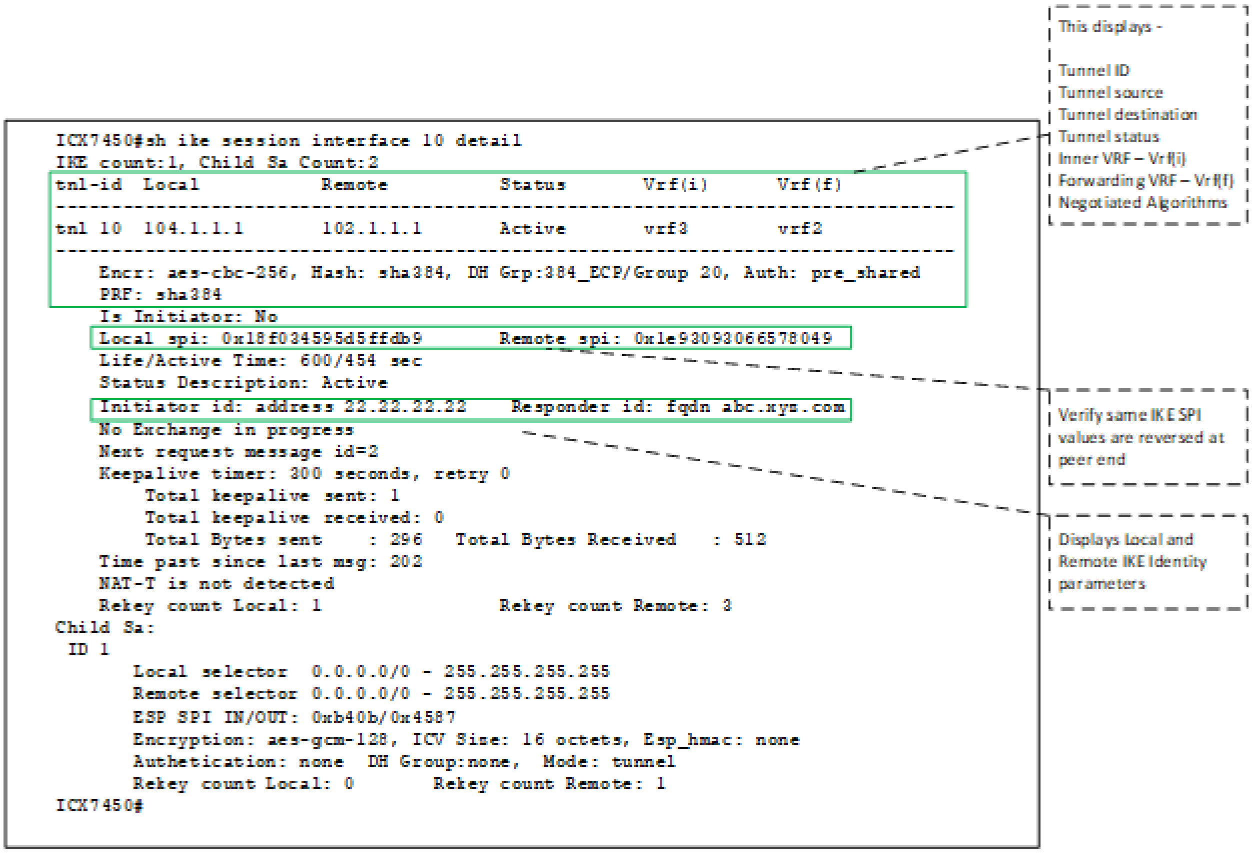
FIGURE 16 Show CPU



Verify CPU utilization is within safe limits

```
ICX7450#sh cpu
cpu0:
1 percent busy, from 498 sec ago
1   sec avg:  1 percent busy
5   sec avg:  1 percent busy
60  sec avg:  1 percent busy
300 sec avg:  1 percent busy
cpu1:
0 percent busy, from 498 sec ago
1   sec avg: 0 percent busy
5   sec avg: 0 percent busy
60  sec avg: 0 percent busy
300 sec avg: 0 percent busy
ICX7450#
```

FIGURE 17 Show IKE Session Interface Detail



```
ICX7450#sh ike session interface 10 detail
IKE count:1, Child Sa Count:2
tnl-id  Local           Remote          Status       Vrf(i)      Vrf(f)
--------------------------------------------------------------------------------
tnl 10  104.1.1.1       102.1.1.1       Active       vrf3        vrf2
--------------------------------------------------------------------------------
    Encr: aes-cbc-256, Hash: sha384, DH Grp:384_ECP/Group 20, Auth: pre_shared
    PRF: sha384
    Is Initiator: No
    Local spi: 0x18f034595d5ffdb9       Remote spi: 0x1e93093066578049
    Life/Active Time: 600/454 sec
    Status Description: Active
    Initiator id: address 22.22.22.22    Responder id: fqdn abc.xys.com
    No Exchange in progress
    Next request message id=2
    Keepalive timer: 300 seconds, retry 0
        Total keepalive sent: 1
        Total keepalive received: 0
        Total Bytes sent    : 296   Total Bytes Received   : 512
    Time past since last msg: 202
    NAT-T is not detected
    Rekey count Local: 1                Rekey count Remote: 3
Child Sa:
 ID 1
       Local selector  0.0.0.0/0 - 255.255.255.255
       Remote selector 0.0.0.0/0 - 255.255.255.255
       ESP SPI IN/OUT: 0xb40b/0x4587
       Encryption: aes-gcm-128, ICV Size: 16 octets, Esp_hmac: none
       Authetication: none  DH Group:none,  Mode: tunnel
       Rekey count Local: 0       Rekey count Remote: 1
ICX7450#
```

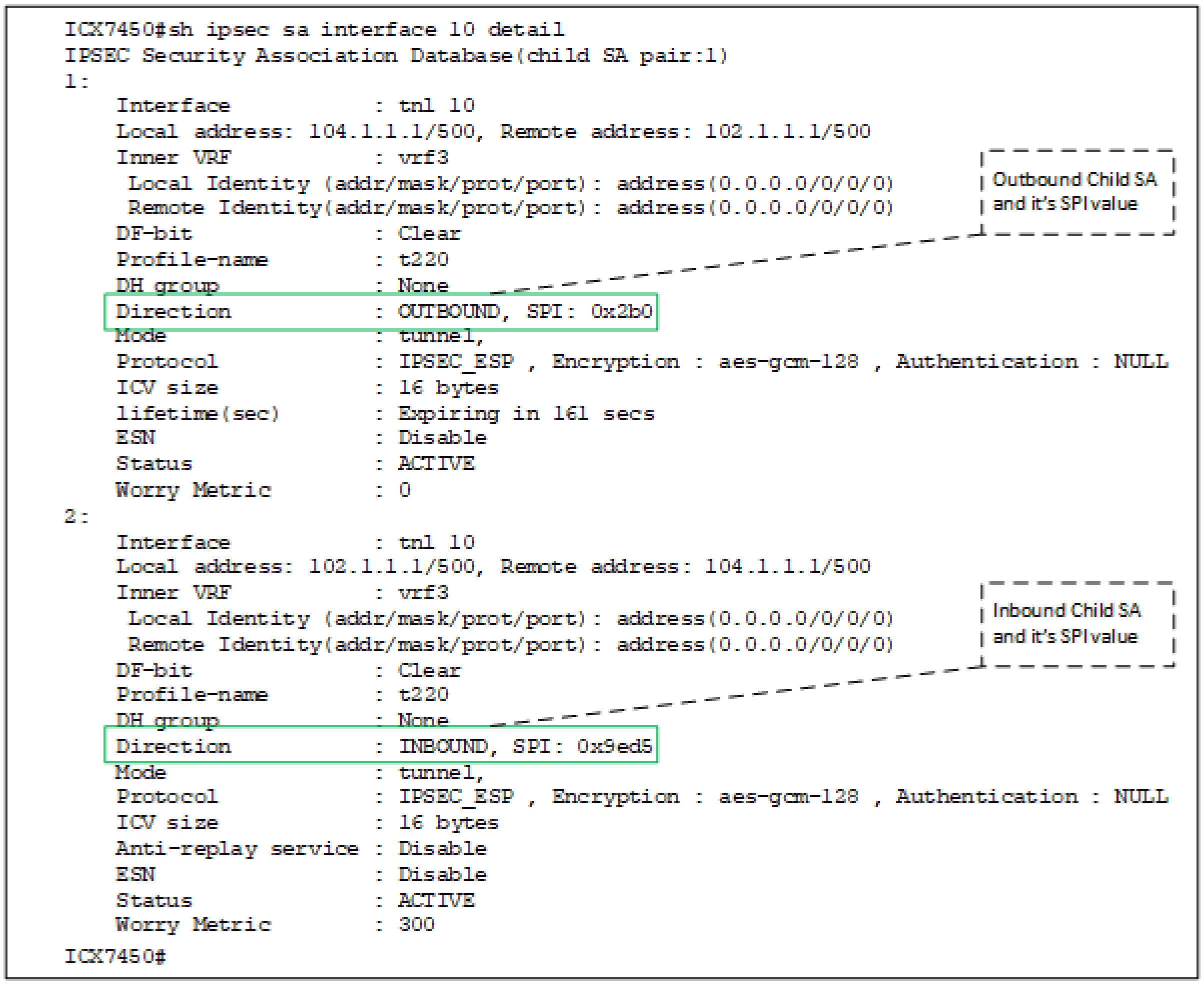
This displays -

Tunnel ID
Tunnel source
Tunnel destination
Tunnel status
Inner VRF – Vrf(i)
Forwarding VRF – Vrf(f)
Negotiated Algorithms

Verify same IKE SPI values are reversed at peer end

Displays Local and Remote IKE Identity parameters

FIGURE 18 Show IPsec SA Interface Detail

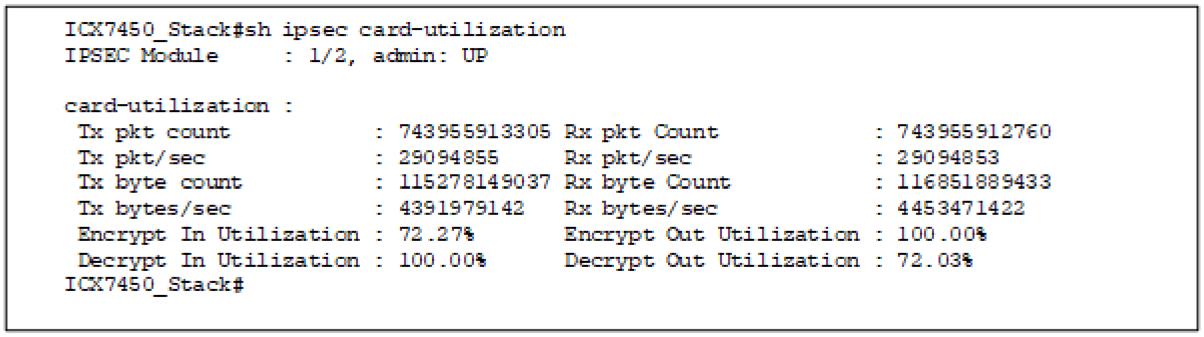


```
ICX7450#sh ipsec sa interface 10 detail
IPSEC Security Association Database(child SA pair:1)
1:
    Interface            : tnl 10
    Local address: 104.1.1.1/500, Remote address: 102.1.1.1/500
    Inner VRF            : vrf3
     Local Identity (addr/mask/prot/port): address(0.0.0.0/0/0/0)
     Remote Identity(addr/mask/prot/port): address(0.0.0.0/0/0/0)
    DF-bit               : Clear
    Profile-name         : t220
    DH group             : None
    Direction            : OUTBOUND, SPI: 0x2b0
    Mode                 : tunnel,
    Protocol             : IPSEC_ESP , Encryption : aes-gcm-128 , Authentication : NULL
    ICV size             : 16 bytes
    lifetime(sec)        : Expiring in 161 secs
    ESN                  : Disable
    Status               : ACTIVE
    Worry Metric         : 0
2:
    Interface            : tnl 10
    Local address: 102.1.1.1/500, Remote address: 104.1.1.1/500
    Inner VRF            : vrf3
     Local Identity (addr/mask/prot/port): address(0.0.0.0/0/0/0)
     Remote Identity(addr/mask/prot/port): address(0.0.0.0/0/0/0)
    DF-bit               : Clear
    Profile-name         : t220
    DH group             : None
    Direction            : INBOUND, SPI: 0x9ed5
    Mode                 : tunnel,
    Protocol             : IPSEC_ESP , Encryption : aes-gcm-128 , Authentication : NULL
    ICV size             : 16 bytes
    Anti-replay service  : Disable
    ESN                  : Disable
    Status               : ACTIVE
    Worry Metric         : 300
ICX7450#
```

Outbound Child SA
and it's SPI value

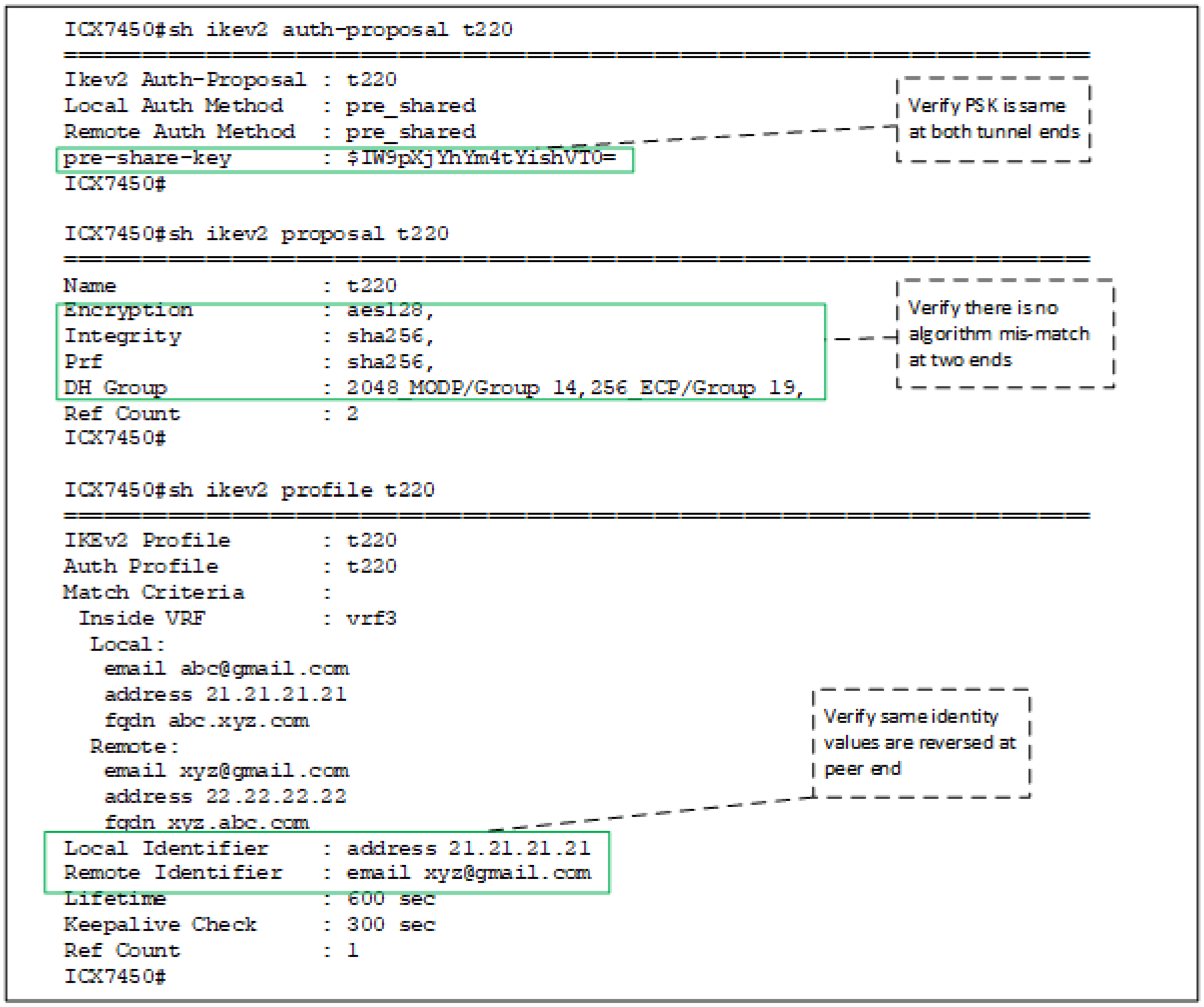
Inbound Child SA
and it's SPI value

FIGURE 19 Show IPsec Card-Utilization



```
ICX7450_Stack#sh ipsec card-utilization
IPSEC Module      : 1/2, admin: UP

card-utilization :
 Tx pkt count             : 743955913305 Rx pkt Count              : 743955912760
 Tx pkt/sec               : 29094855     Rx pkt/sec                : 29094853
 Tx byte count            : 115278149037 Rx byte Count             : 116851889433
 Tx bytes/sec             : 4391979142   Rx bytes/sec              : 4453471422
 Encrypt In Utilization : 72.27%       Encrypt Out Utilization : 100.00%
 Decrypt In Utilization : 100.00%      Decrypt Out Utilization : 72.03%
ICX7450_Stack#
```

FIGURE 20 Checking IKEv2 Parameter Settings



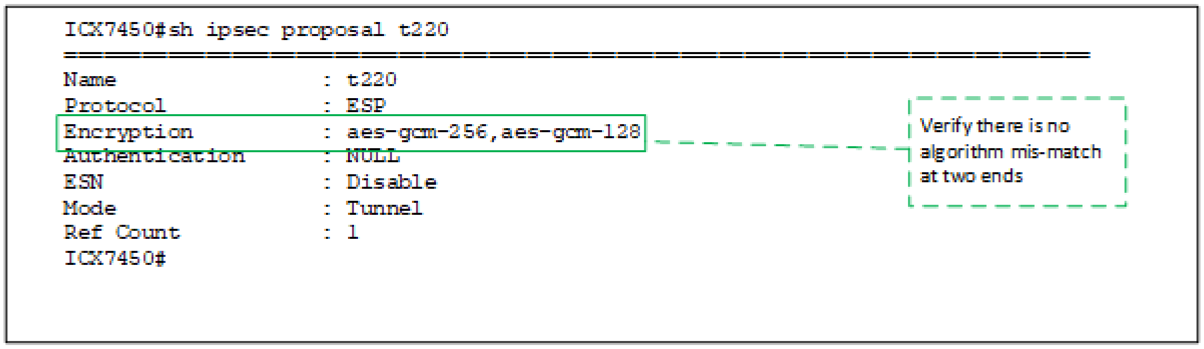
```
ICX7450#sh ikev2 auth-proposal t220
=================================================================================
Ikev2 Auth-Proposal : t220
Local Auth Method   : pre_shared
Remote Auth Method  : pre_shared
pre-share-key       : $IW9pXjYhYm4tYishVT0=
ICX7450#

ICX7450#sh ikev2 proposal t220
=================================================================================
Name                    : t220
Encryption              : aes128,
Integrity               : sha256,
Prf                     : sha256,
DH Group                : 2048_MODP/Group 14,256_ECP/Group 19,
Ref Count               : 2
ICX7450#

ICX7450#sh ikev2 profile t220
=================================================================================
IKEv2 Profile           : t220
Auth Profile            : t220
Match Criteria          :
  Inside VRF            : vrf3
   Local:
     email abc@gmail.com
     address 21.21.21.21
     fqdn abc.xyz.com
   Remote:
     email xyz@gmail.com
     address 22.22.22.22
     fqdn xyz.abc.com
Local Identifier        : address 21.21.21.21
Remote Identifier       : email xyz@gmail.com
Lifetime                : 600 sec
Keepalive Check         : 300 sec
Ref Count               : 1
ICX7450#
```

Verify PSK is same at both tunnel ends

Verify there is no algorithm mis-match at two ends

Verify same identity values are reversed at peer end

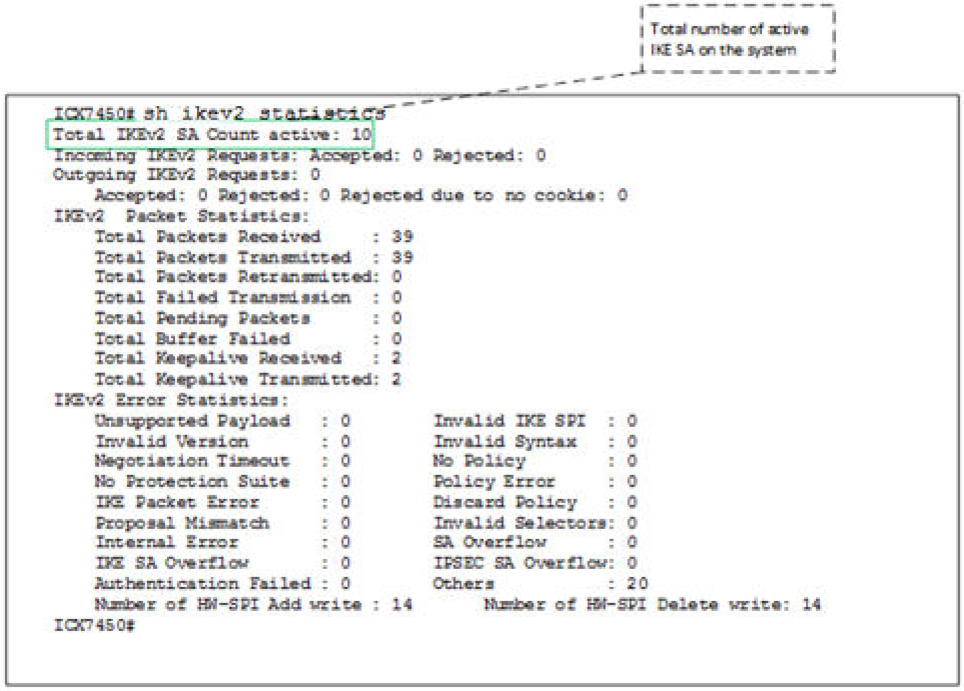
FIGURE 21 Show IPsec Proposal



```
ICX7450#sh ipsec proposal t220
=================================================================================
Name                    : t220
Protocol                : ESP
Encryption              : aes-gcm-256,aes-gcm-128
Authentication          : NULL
ESN                     : Disable
Mode                    : Tunnel
Ref Count               : 1
ICX7450#
```

Verify there is no algorithm mis-match at two ends

FIGURE 22 Show IKEv2 Statistics



Total number of active IKE SA on the system

```
ICX7450# sh ikev2 statistics
Total IKEv2 SA Count active: 10
Incoming IKEv2 Requests: Accepted: 0 Rejected: 0
Outgoing IKEv2 Requests: 0
    Accepted: 0 Rejected: 0 Rejected due to no cookie: 0
IKEv2  Packet Statistics:
    Total Packets Received     : 39
    Total Packets Transmitted  : 39
    Total Packets Retransmitted: 0
    Total Failed Transmission  : 0
    Total Pending Packets      : 0
    Total Buffer Failed        : 0
    Total Keepalive Received   : 2
    Total Keepalive Transmitted: 2
IKEv2 Error Statistics:
    Unsupported Payload    : 0        Invalid IKE SPI   : 0
    Invalid Version        : 0        Invalid Syntax    : 0
    Negotiation Timeout    : 0        No Policy         : 0
    No Protection Suite    : 0        Policy Error      : 0
    IKE Packet Error       : 0        Discard Policy    : 0
    Proposal Mismatch      : 0        Invalid Selectors: 0
    Internal Error         : 0        SA Overflow       : 0
    IKE SA Overflow        : 0        IPSEC SA Overflow: 0
    Authentication Failed : 0         Others            : 20
    Number of HW-SPI Add write : 14        Number of HW-SPI Delete write: 14
ICX7450#
```

Debugging

Tunnel Fails to Come Up

TABLE 3 Tunnel Fails to Come Up

Step	Action	Reference
1	Verify IPsec module status.	Refer to Figure 12 on page 37.
	If IPsec module status is not OK.	Check if IPsec card is seated properly. If yes, contact TAC for further assistance.
2	Verify IKEv2 SA status.	Refer to Figure 14 on page 38.
	If IKEv2 SA is not initiated or is in constructing state.	Check Layer 3 reachability between tunnel source and destination.
		If Layer 3 reachability is OK, refer to Figure 15 on page 38.
3	If IKEv2 SA status is OK, verify IPsec SA status.	Refer to Figure 15 on page 38.
	If IPsec SA is down.	Refer to Figure 22 on page 42.
4	If both IKEv2 SA and IPsec SA status is OK.	Refer to Figure 16 on page 38, Figure 17 on page 39, and Figure 18 on page 40.
		Contact TAC for further assistance.

Tunnel Is Up But Traffic Forwarding Failing

Collect information according to Figure 12 on page 37 and Figure 14 on page 38. Contact TAC for further assistance.

Commonly Used Debug and DM Commands

Debug Commands

debug ikev2 { **all** | **error** | **event** | **packet** [**detail** | **receive** | **send**] | **peer** *ip-address* | **trace** [**info**] }

> **NOTE**
> Using the **debug ikev2 all** command is required after enabling the specific debug filters to start debugging.

DM Commands

1. The following commands must be used only on the stack unit where the ICX7400-SERVICE-MOD module resides:
   - **dm ipsec all-stat**
   - **dm ip-port ipv4 tunnel-info** *tnl-id*
   - **dm ipsec card-info**
   - **dm ipsec pp tunnel-verify** *tnl-id*
   - **dm ipsec module** *unit/module* **stats brief**
   - **dm ipsec module** *unit/module* **stats**
   - **dm ipsec module** *unit/module* **error-count**
2. The following commands must be used on all the units of the stack system:
   - **dm ipsec cpu err-buf show**
   - **dm ipv4-unicast hw-route** (if tunnel base path is in def-vrf)
   - **dm ipv4-unicast vrf** *vrf-name* **hw-route** (if tunnel base path is in user-vrf)
   - **dm ip-nexthop usage all**
   - **dm ipv6-unicast hw-route** (for IPv6 tunnel debugging)

IPsec Scalability Limits

The ICX 7450 supports encryption and decryption on the ICX7400-SERVICE-MOD module. Encryption and decryption are hardware-based and are not performed by the software.

One ICX7400-SERVICE-MOD module can be installed per device or per stack. The ICX7400-SERVICE-MOD module supports a maximum of 100 IPsec tunnels. If you try to configure more than 100 tunnels, an error message is displayed. To configure a new IPsec tunnel when the maximum number allowed is already configured, you must remove an existing tunnel. Limits also apply to the elements that are used to configure IPsec tunnels. The element thresholds are set out in the following table.

TABLE 4 IPsec Tunnel Element Thresholds

Tunnel Element	Threshold
IKEv2 Proposal	128
IKEv2 Auth Proposal	128
IKEv2 Policy	128
IKEv2 Profile	128
IPsec Proposal	128
IPsec Profile	128
IPsec Tunnels	100 (either IPv4 or IPv6 or together)

Downgrade Considerations

When the system software is downgraded to a version that does not support IPsec, an error message is displayed. To prevent the display of error messages after a downgrade, remove any IPsec-related configurations before the downgrade. Software versions that do not support IPsec do not recognize the ICX7400-SERVICE-MOD module.

RUCKUS™
an ARRIS company

Ruckus Wireless, Inc., a wholly owned subsidiary of ARRIS International plc.
350 West Java Dr., Sunnyvale, CA 94089 USA
www.ruckuswireless.com