# Resource - ipsectunnel

## About this Resource

Defines the IPsec Tunnel Managed Object

## REST API Methods for this Resource

- GET ipsectunnel
- GET ipsectunnel id
- POST ipsectunnel id
- PUT ipsectunnel id
- DELETE ipsectunnel id

## Resource Schema

### Configuration

| Parameter Name | Required | Service Affecting | Data Type | Default Value | Possible Values | Description |
|---|---|---|---|---|---|---|
| **TunnelActivation** | Yes | Yes | Enum | 1 | Possible values:<br>• 0 - eAlways<br>• 1 - eLinkMonitorAction | Activates SBC communication with remote IPsec peer by intiating the IKE and IPsec phase negotiations as permanent or on-demand service type. This parameter is applicable when the "Operating Mode" is set to "Initiator".<br>• Always: Always initiates the IKE Security Association(SA) and IPsec phase negotiations permanently with the remote IPsec peer.<br>• Link Monitor Action: Initiates the IKE and IPsec phase negotiations with the remote IPsec peer as on-demand upon request from the link monitor switch-over action. |
| **TunnelName** | Yes | Yes | string | none | 64 - Max Length | Specifies the IPsec tunnel name that this IPsec object is associated with. The tunnel name must not contain any space characters. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **OperatingMode** | Yes | Yes | Enum | 0 | Possible values:<br>• 0 - eInitiator<br>• 1 - eResponder | Controls SBC communication with remote peer for IKE negotiations and IPsec connections.<br>• Initiator mode: Enables the branch office SBC gateway to initiate the IKE Security Association(SA) and IPsec tunnel negotiation request.<br>• Responder mode: Enables the corporate SBC gateway to receive the request to establish an IKE/IPsec tunnel connection. |
| **allowAnyLocalAddress** | Yes | Yes | Enum | 0 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | • True: signifies the local address to be filled in during negotiation by automatic keying although a concrete local address has been assigned.<br>• False: signifies that the static Local Address assignment parameter is used. |
| **localAddress** | Yes | Yes | string | none | 255 - Max Length | Specifies the IP address or fully-qualified domain name of the local network interface. If "Allow any address" is set True, then it will allow any outgoing address during negotiations. |
| **allowAnyRemoteAddress** | Yes | Yes | Enum | 0 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | • True: signifies the remote address to be filled in during negotiation by automatic keying although a concrete remote address has been assigned.<br>• False: signifies that the static Remote Address assignment parameter is used. |
| **remoteAddress** | Yes | Yes | string | none | 255 - Max Length | Specifies the IP address or fully-qualified domain name of the remote network interface. If "Allow any address" is set True, then it will allow any incoming address during negotiations. |
| **localSubnetAddress** | Yes | Yes | string | none | 200 - Max Length | Specifies the IP address of the private subnet behind the local network interface. This can be expressed as network/netmask. Maximum of 10 subnets can be specified by separated commas. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **remoteSubnetAddress** | Yes | Yes | string | none | 200 - Max Length | Specifies the IP address of the private subnet behind the remote network interface. This can be expressed as network/netmask. Maximum of 10 subnets can be specified by separated commas. |
| **applyPolicyRules** | Yes | No | Enum | 1 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | • True: signifies that the local gateway is doing forwarding-firewalling using iptables for traffic from Local Subnet Address and Remote Subnet Address.<br>• False: signifies that the iptables policy rules are not created for traffic to and from the peer endpoint. |
| **useSANIdentifier** | No | No | Enum | 0 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | • True: The Subject Alternative Name(SAN) identifier must be configured in the "localSANIdentifier" attribute and sent to the remote gateway for an authentication config match.<br>• False: By default, the SBC Certificate's Subject Distinguished Name(Subject DN) identifier is automatically extracted from the certificate and sent to the remote gateway for an authentication config match. |
| **localSANIdentifier** | Yes | Yes | string | none | 255 - Max Length | Specifies the configured Subject Alternative Name(SAN) identifier to be sent to the remote gateway for a peer authentication config match. If "peerAuthIdentifier" on the remote gateway is configured to authenticate a SAN identifier from the peer's certificate, it will attempt to match its configured SAN identifier with the expected SAN identifier retrieved from the peer authentication config. If "useSANIdentifier" is set True, the Subject Alternative Name(SAN) identifier must be picked from a list of DNS names displayed under the local attributes for the 'SBC Certificate'.<br>ⓘ This option is available if "UseSANIdentifier" is set to "True". |

| | | | | | | |
|---|---|---|---|---|---|---|
| **peerAuthMode** | Yes | Yes | Enum | 0 | Possible values:<br>• 0 - eAuthCertificate<br>• 1 - eAuthPresharedKey | Specifies the authentication method required from the remote side. Certificate authentication mode: Specifies the use of public key signature when authenticating the peer IPsec gateway. The system must contain server certificate/private key, Certificate Authority(CA) which signed the certificate and peers CA for identifying the peer. Preshared Key authentication mode: Specifies the key to be shared with the peer. This key must match the same key configured on the peer system. |
| **peerAuthIdentifier** | Yes | Yes | string | none | 255 - Max Length | Specifies how the peer should be identified for IKE certificate authentication. On selection of Certificate as the peer authentication mode, valid identifier should be set to the peer certificate's Subject Alternative Name(SAN) or the peer's subject Distinguished Name (Subject DN). Alternatively, if SAN or Subject DN is not known, it can be configured for 'any' on the SBC responder-side gateway configured for 'any' remote address.<br>ⓘ This option is available if "Peer Authentication Mode" is set to "Certificate". |
| **remoteIdentifier** | Yes | Yes | string | none | 255 - Max Length | Specifies how the peer should be identified for IKE preshared key authentication. The identifier selector can be an all host address(0.0.0.0), a specific IP address or a fully-qualified domain name of the remote LAN network interface.<br>ⓘ This option is available when "Allow any address" is set "True" and the "Peer Authentication Mode" is set to "Preshared Key". |

| | | | | | | |
|---|---|---|---|---|---|---|
| **EncryptedPresharedKey** | Yes | Yes | string | none | 256 - Max Length | Specifies the secret value which is shared with the peer. On selection of Preshared Key as the peer authentication mode, the secret value can be a pass-phrase or hex string. This key must match the key configured on the peer system.<br>This option is available if "Peer Authentication Mode" is set to "Preshared Key". |
| **encryption** | Yes | No | Enum | 1 | Possible values:<br>• 0 - aes256<br>• 1 - aes128<br>• 2 - des_cbc3 | The Internet Key Exchange(IKE) protocol establishes a secure channel for IKE Phase 1 protected authentication and IPsec Phase 2 traffic protection with the Encapsulating Security Payload(ESP) protocol using the encryption algorithm to provide data confidentiality. |
| **integrity** | Yes | No | Enum | 0 | Possible values:<br>• 0 - sha1<br>• 1 - sha256 | The Internet Key Exchange(IKE) protocol establishes a secure channel for IKE Phase 1 protected authentication and IPsec Phase 2 traffic protection with the Encapsulating Security Payload(ESP) protocol using the hash algorithm to provide integrity. |
| **dhgroup** | Yes | No | Enum | 3 | Possible values:<br>• 0 - dhgroup1<br>• 1 - dhgroup2<br>• 2 - dhgroup5<br>• 3 - dhgroup14 | The Internet Key Exchange(IKE) protocol establishes a secure channel for IKE Phase 1 protected authentication and IPsec Phase 2 traffic protection with the Encapsulating Security Payload(ESP) protocol using the encryption algorithm to provide authenticity. |
| **enablePFS** | Yes | No | Enum | 1 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | If enabled, a new ISAKMP SA is created for each IPsec SA negotiation and a Diffie-Hellman exchange is performed for each IPsec SA negotiation. True: signifies that DH Group is defined by the Phase 2 Diffie-Hellman Group parameter. False: signifies that Phase 2 DH Group will automatically not be specified and exchanged for IPsec Phase 2 negotiations. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **enableRekeying** | Yes | No | Enum | 0 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | True: Initiate SA Negotiation upon connection expiry. Applies to both IKE SA and IPsec SA. False: SA Negotiation is not initiated upon connection expiry. |
| **enableReauthentication** | Yes | No | Enum | 0 | Possible values:<br>• 0 - btFalse<br>• 1 - btTrue | • True: IKE SA Rekeying also initiates Peer Authentication. IKE and IPsec SA's are uninstalled then recreated.<br>• False: IKE SA Rekeying performed without the Peer Authentication |
| **keyingRetries** | Yes | No | int | 3 | Possible values:<br>• 1 - Minimum<br>• 10 - Maximum | Specifies how many attempts should be made to negotiate a connection. This parameter applies to both IKE SA and IPsec SA. |
| **ikeLifetime** | Yes | No | int | 10800 | Possible values:<br>• 3600 - Minimum<br>• 86400 - Maximum | Specifies the life time of IKE SA connection, from successful negotiation to expiry. |
| **ipsecLifetime** | Yes | No | int | 3600 | Possible values:<br>• 3600 - Minimum<br>• 86400 - Maximum | Specifies the life time of IPsec SA connection, from successful negotiation to expiry. |
| **marginTime** | Yes | No | int | 600 | Possible values:<br>• 60 - Minimum<br>• 600 - Maximum | Time before SA expiry the rekeying should start. Applies to both IKE SA and IPsec SA. |