Shared Use of IPsec Tunnel in a Multi-VPN Environment

draft-he-ipsecme-vpn-shared-ipsecsa

Qi He
Wei Pan
Xiaolan Chen
Beijing Ding

IETF 119
March 2024
Background

- Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space
• Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space

• When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2
  • If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.
### Background

- Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space
- When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2
  - If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.
  - If VPN 1 and VPN 2 separately use different IKE SAs, when negotiating the creation of Child SA, the receiver can differentiate which VPN this Child SA should be associated with.

<table>
<thead>
<tr>
<th>Device A</th>
<th>Device B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VPN 1 (10.1.1.0/24)</strong></td>
<td><strong>VPN 1 (11.1.1.0/24)</strong></td>
</tr>
<tr>
<td><strong>VPN 2 (10.1.1.0/24)</strong></td>
<td><strong>VPN 2 (11.1.1.0/24)</strong></td>
</tr>
</tbody>
</table>

**IKE SA 1**

CREATE_CHILD_SA request

(TS_i((0,0,10.1.1.0-10.1.1.255)), \rightarrow) \rightarrow

(TS_r((0,0,11.1.1.0-11.1.1.255)))

This Child SA belongs to VPN 1, because its IKE SA belongs to VPN 1

**IKE SA 2**

CREATE_CHILD_SA request

(TS_i((0,0,10.1.1.0-10.1.1.255)), \rightarrow)

(TS_r((0,0,11.1.1.0-11.1.1.255)))

This Child SA belongs to VPN 2, because its IKE SA belongs to VPN 2
Background

- Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space

- When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2
  - If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.
  - If VPN 1 and VPN 2 separately use different IKE SAs, when negotiating the creation of Child SA, the receiver can differentiate which VPN this Child SA should be associated with.

- Therefore, currently, different VPNs need different IPsec tunnels (different IKE SAs & Child SAs)
Problem Statement

• In 3GPP networks, full-meshed IPsec tunnels are established among base stations.

Base Station A

IPsec Tunnel 1

Base Station B

IPsec Tunnel 2

Base Station C

IPsec Tunnel 5

Base Station D

IPsec Tunnel 3

IPsec Tunnel 6

• Radio Access Network (RAN) Sharing is used to lease the infrastructure to other operators.

Base Station A

VPN 1 (Operator A)

VPN 2 (Operator B)

VPN N (Operator N)

Base Station B

VPN 1 (Operator A)

VPN 2 (Operator B)

VPN N (Operator N)

IPsec Tunnel 1

IPsec Tunnel 2

IPsec Tunnel N

• IPsec tunnels’ number is seriously boosted as the number of base stations and operators sharing the RAN increases.
  • Assume there are \( N \) neighbors and \( M \) sharing operators, then the IKE SAs are \( N \times M \) and the Child SAs are at least \( N \times M \).

• The limited SAs supported by the device restricts the development and evolution of services in this scenario.
Solution Overview

• **Core Concept:** Share the same IPsec tunnel for different VPNs, by adding VPN-related information in the creation of Child SA and the IPsec data packets

  - **Step 1**
    - Negotiation of Support in IKE_SA_INIT
  - **Step 2**
    - Correlate VPN with Child SA during its creation
  - **Step 3**
    - Carry VPN info in the IPsec data packets

• **Current Design:**
  - Add the VPN attribute for each Traffic Selector when negotiating the traffic to be protected in Child SAs.
  - Carry the VPN info in the extended ESP and AH header to distinguish which VPN the inner packet belongs to.
Solution Step 1

- During the IKE_SA_INIT exchange, two peers negotiate the support of correlating VPN with IPsec SAs.

- Peers include the **VPN_BASED_TS_SUPPORTED** notify payload in the IKE_SA_INIT exchange request and response, to indicate the support of using new Traffic Selectors that contain the VPN ID field.

```
IKE_SA_INIT Message Exchange Example

Initiator                         Responder
-----------------------------------------------
HDR, SAi1, KEi, Ni,
     N(VPN_BASED_TS_SUPPORTED) -->
     N(VPN_BASED_TS_SUPPORTED)
```

```
<-- HDR, SAr1, KEr, Nr, [CERTREQ,]
     N(VPN_BASED_TS_SUPPORTED)
```
Solution Step 2

- Two New Traffic Selectors are introduced: `TS_IPV4_ADDR_RANGE_VPN` and `TS_IPV6_ADDR_RANGE_VPN`
  - Compared with existing v4/v6 Traffic Selectors, these two new Traffic Selectors contain an additional "VPN ID" field.

- When creating Child SAs, two peers using these two new Traffic Selectors instead of the existing two.
  - Parsing Rule: First pairing the Traffic Selectors with the same VPN ID from the TSi and TSr payloads, then processing the paired Traffic Selectors.

### CREATE_CHILD_SA Message Exchange Example

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK {SA, Ni, [KEi,}</td>
<td>HDR, SK {SA, Nr, [KEr,}</td>
</tr>
<tr>
<td><code>TSi(TS_IPV4_ADDR_RANGE_VPN),</code></td>
<td><code>TSr(TS_IPV4_ADDR_RANGE_VPN)</code></td>
</tr>
<tr>
<td><code>TSr(TS_IPV4_ADDR_RANGE_VPN)</code></td>
<td><code>--&gt;</code></td>
</tr>
<tr>
<td>HDR, SK {SA, Ni, [KEi,}</td>
<td>HDR, SK {SA, Nr, [KEr,}</td>
</tr>
<tr>
<td><code>TSi(TS_IPV6_ADDR_RANGE_VPN),</code></td>
<td><code>TSr(TS_IPV6_ADDR_RANGE_VPN)</code></td>
</tr>
<tr>
<td><code>TSr(TS_IPV6_ADDR_RANGE_VPN)</code></td>
<td><code>--&gt;</code></td>
</tr>
</tbody>
</table>

### TS_IPV4_ADDR_RANGE_VPN and TS_IPV6_ADDR_RANGE_VPN Formats

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS Type</td>
<td>Traffic Selector Type</td>
<td>1 2</td>
</tr>
<tr>
<td>IP Protocol ID*</td>
<td>IP Protocol Identification</td>
<td>3</td>
</tr>
<tr>
<td>Selector Length</td>
<td>Length of Selector</td>
<td>4 5 6 7 8</td>
</tr>
<tr>
<td>Start Port*</td>
<td>Starting Port of Selector</td>
<td>2 3 4 5</td>
</tr>
<tr>
<td>End Port*</td>
<td>Ending Port of Selector</td>
<td>2 3 4 5</td>
</tr>
<tr>
<td>Starting Address*</td>
<td>Starting Address of Selector</td>
<td>8 9 0 1</td>
</tr>
<tr>
<td>Ending Address*</td>
<td>Ending Address of Selector</td>
<td>8 9 0 1</td>
</tr>
<tr>
<td>VPN ID</td>
<td>VPN ID</td>
<td>9</td>
</tr>
</tbody>
</table>
Solution Step 3

- Extending the ESP and AH packet formats with an additional "VPN ID" field, to differentiate which VPN the inner traffic belongs to.
Alternative Solutions

• Splitting the 32-bit SPI into two sub-fields: the VPN ID sub-field and SPI sub-field

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPN ID</td>
<td>SPI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• When creating Child SAs, to set the VPN ID sub-field all zero and only use the SPI sub-field
• When sending IPsec packets, to set the VPN ID sub-field with the actual VPN ID value that the inner traffic belongs to, and to set the SPI sub-field with the SPI value

• Advantage
  • No ESP/AH packet format changes needed

• Disadvantage
  • Scalable issue: 16-bit VPN ID is needed for future scenarios, then 16-bit SPI might not be sufficient.
  • Packet disorder: Different VPNs use different 32-bit SPI (composed of VPN ID and actual SPI) in the data packets, this will interfere with the load balance process of the on-path routers who look at the SPI field when doing the hash, and finally cause disorder at the receiver.

• Using a notify or a traffic selector of just the VPN ID when creating the Child SAs
  • Can’t differentiate which v4/v6 Traffic Selector is associated with which VPN.
  • May cause unwanted traffic to be included.
Further Considerations

• Is this problem worth solving?

• Suggestions, comments, reviews, co-authors, etc., are all welcome.