Abstract

This draft defines methods to encapsulate and decapsulate ESP packets inside UDP packets for the purpose of traversing NATs.

ESP encapsulation as defined in this document is capable of being used in both IPv4 and IPv6 scenarios.

The encapsulation is used whenever negotiated using IKE, as defined in [Kiv00], or another key management protocol. The design choices are documented in [Dixon00].

Change Log

Version -01
- removed everything related to the AH-protocol
- added instructions on how to use the encapsulation with some other key management protocol than IKE

1. Introduction
It is up to the need of the clients whether transport mode or tunnel mode is to be supported. L2TP/IPsec clients MUST support transport mode, and IPsec tunnel mode clients MUST support tunnel mode.

An IKE implementation supporting this draft MUST NOT generate packets where the Initiator Cookie field is all zeroes. This ensures that IKE packets and ESP packets can be distinguished from each other.

Usage with another key management protocol is described in a separate section.

ESP encapsulation as defined in this document is capable of being used in both IPv4 and IPv6 scenarios.

2. Packet Formats

2.1 UDP-encapsulated ESP Header Format

```
| 0                   1                   2                   3 |
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1                |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Source Port | Destination Port |                             |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Length | Checksum |                             |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Non-IKE Marker |                       |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Non-IKE Marker |                       |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Non-IKE Marker |                       |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Non-IKE Marker |                       |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| ~                      ESP header [RFC 2406] | ~                     |
| ~                                                               |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
```

The UDP header is a standard [RFC 768] header, where
- Source Port and Destination Port are the same as used by IKE traffic.
- Checksum is zero.

Non-IKE Marker is 8 bytes of zero aligning with the Initiator Cookie of an IKE packet.

2.3 NAT-keepalive Packet Format

```
| 0                   1                   2                   3 |
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1                |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Source Port | Destination Port |                             |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | Length | Checksum |                             |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
| | OxFF |                   |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
```

The UDP header is a standard [RFC 768] header, where
- Source Port and Destination Port are the same as used by IKE traffic.
- Checksum is zero.
The sender SHOULD use a one octet long payload with the value 0xFF. The receiver SHOULD ignore a received NAT-keepalive packet.

3. Encapsulation and Decapsulation Procedures

3.1 Auxiliary Procedures

3.1.1 Tunnel Mode Decapsulation NAT Procedure

When a tunnel mode has been used to transmit packets, the inner IP header can contain addresses that are not suitable for the current network. This procedure defines how these addresses are to be converted to suitable addresses for the current network.

Depending on local policy, one of the following MUST be done:

a) If a valid source IP address space has been defined in the policy for the encapsulated packets from the peer, check that the source IP address of the inner packet is valid according to the policy.

b) If an address has been assigned for the remote peer, check that the source IP address used in the inner packet is the same as the IP address assigned.

c) NAT is performed for the packet, making it suitable for transport in the local network.

3.1.2 Transport Mode Decapsulation NAT Procedure

When a transport mode has been used to transmit packets, contained TCP or UDP headers will contain incorrect checksums due to the change of parts of the IP header during transit. This procedure defines how to fix these checksums.

Depending on local policy, one of the following MUST be done:

a) If the protocol header after the ESP header is a TCP/UDP header, zero the checksum field in the TCP/UDP header.

b) If the protocol header after the ESP header is a TCP/UDP header, recompute the checksum field in the TCP/UDP header.

c) If the protocol header after the ESP header is a TCP/UDP header and the peer’s real source IP address has been received according to [Kiv00], incrementally recompute the TCP/UDP checksum:
   - subtract the IP source address in the received packet from the checksum
   - add the real IP source address received via IKE to the checksum

In addition an implementation MAY fix any contained protocols that have been broken by NAT.

3.2 Transport Mode ESP Encapsulation

BEFORE APPLYING ESP/UDP
-------------------------------
IPv4  | orig IP hdr | (any options) | TCP | Data |
---    |-------------|---------------|-----|------|
AFTER APPLYING ESP/UDP
--------------------------------------------------------------------------
IPv4  | orig IP hdr | UDP | Non-ESP | ESP | ESP | ESP |
      | (any options)| Hdr | IKE | Hdr | TCP | Data | Trailer | Auth |
--------------------------------------------------------------------------
|<----- encrypted ---->|
1) Ordinary ESP encapsulation procedure is used.
2) A properly formatted UDP header and a Non-IKE Marker are inserted where shown.
3) The Total Length, Protocol and Header Checksum fields in the IP header are edited to match the resulting IP packet.

3.3 Transport Mode ESP Decapsulation

1) The UDP header and the Non-IKE Marker are removed from the packet.
2) The Total Length, Protocol and Header Checksum fields in the new IP header are edited to match the resulting IP packet.
3) Ordinary ESP decapsulation procedure is used.
4) Transport mode decapsulation NAT procedure is used.

3.4 Tunnel Mode ESP Encapsulation

BEFORE APPLYING ESP/UDP

IPv4 | orig IP hdr |     |     |
| (any options) | TCP | Data |

AFTER APPLYING ESP/UDP

IPv4 | new h. | UDP | Non- | ESP | orig IP hdr |     |     |     |
| (opts) | Hdr | IKE | Hdr | (any options) | TCP | Data | Trail | Auth |

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-------- encrypted --------&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---------------- authenticated ----------------&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) Ordinary ESP encapsulation procedure is used.
2) A properly formatted UDP header and a Non-IKE Marker are inserted where shown.
3) The Total Length, Protocol and Header Checksum fields in the new IP header are edited to match the resulting IP packet.

3.5 Tunnel Mode ESP Decapsulation

1) The UDP header and the Non-IKE Marker are removed from the packet.
2) The Total Length, Protocol and Header Checksum fields in the new IP header are edited to match the resulting IP packet.
3) Ordinary ESP decapsulation procedure is used.
4) Tunnel mode decapsulation NAT procedure is used.

4. NAT Keepalive Procedure

The sole purpose of sending NAT-keepalive packets is to keep NAT mappings alive for the duration of a connection between the peers. Reception of NAT-keepalive packets MUST NOT be used to detect liveness of a connection.

A peer MAY send a NAT-keepalive packet if there exists one or more phase I or phase II SAs between the peers, or such an SA has existed at most N minutes earlier. N is a locally configurable parameter with a default value of 5 minutes.
A peer SHOULD send a NAT-keepalive packet if a need to send such packets is detected according to [Kiv00] and if no other packet to the peer has been sent in M seconds. M is a locally configurable parameter with a default value of 20 seconds.

5. Usage with Another Key Management Protocol

5.1. Requirements

The important requirements when using the encapsulation method with another key management protocol are:

a) It must be possible to distinguish key management packets from ESP packets.

b) If more than one UDP port pair is being used, all the relevant NAT mappings must be kept alive.

5.2. Alternative Encapsulation Method 1 - Common Port

IPv4 | IP hdr | UDP | ESP | ...ESP packet...
     | (options) | Hdr | Hdr |

IPv4 | IP hdr | UDP | Non-ESP marker | ...key management packet...
     | (options) | Hdr | ESP |

Non-ESP marker in this case is 4 bytes of zero. The same port pair is used for both types of traffic, and the keepalive mechanism is as defined in this document for IKE traffic. It is required that an implementation using this method does not use ESP SPIs that are equal to zero.

This method is more efficient than the one defined for IKE traffic because it makes the more frequent packets smaller.

5.2. Alternative Encapsulation Method 2 - Separate Ports

IPv4 | IP hdr | UDP | ESP | ...ESP packet...
     | (options) | Hdr | Hdr |

IPv4 | IP hdr | UDP | ...key management packet...
     | (options) | Hdr |

In this method the two types of traffic use different UDP ports, so no non-something markers are needed. Both UDP ports must be kept alive using the keepalive procedure.

Whether or not this results in better bandwidth utilization than using a common UDP port depends on the traffic characteristics. There is less overhead per packet, but more need for keepalive packets.

6. Intellectual Property Rights

The IETF has been notified of intellectual property rights claimed in
regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

SSH Communications Security Corp has notified the working group of one or more patents or patent applications that may be relevant to this internet-draft. SSH Communications Security Corp has already given a licence for those patents to the IETF. For more information consult the online list of claimed rights.

7. Acknowledgments

Thanks to Joern Sierwald, Tamir Zegman, Larry DiBurro, Tatu Ylonen and Santeri Paavolainen who contributed to the previous drafts about NAT traversal.

8. References


[RFC-2119] Bradner, S., "Key words for use in RFCs to indicate Requirement Levels", March 1997


[Dixon00] Dixon, W. et. al., draft-ietf-ipsec-udp-encaps-justification-00.txt, "IPSec over NAT Justification for UDP Encapsulation", June 2001


9. Authors’ Addresses

Ari Huttunen
F-Secure Corporation
Tammasaarenkatu 7
FIN-00181 HELSINKI
Finland
E-mail: Ari.Huttunen@F-Secure.com

William Dixon
Microsoft
One Microsoft Way
Redmond WA 98052
E-mail: wdixon@microsoft.com

Brian Swander
Microsoft
One Microsoft Way
Redmond WA 98052
E-mail: briansw@microsoft.com

Tero Kivinen
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 HELSINKI
Finland
E-mail: kivinen@ssh.fi
Markus Stenberg  
SSH Communications Security Corp  
Fredrikinkatu 42  
FIN-00100 HELSINKI  
Finland  
E-mail: mstenber@ssh.com

Victor Volpe  
Cisco Systems  
124 Grove Street  
Suite 205  
Franklin, MA 02038  
E-mail: vvolpe@cisco.com

Larry DiBurro  
Nortel Networks  
80 Central Street  
Boxborough, MA 01719  
l diburro@nortelnetworks.com