IPSECME Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 September 2025

A. Antony
S. Klassert
secunet

18 March 2025

IKEv2 negotiation for Bound End-to-End Tunnel (BEET) mode ESP
draft-ietf-ipsecme-ikev2-beet-mode-00

Abstract

This document specifies a new Notify Message Type Payload for the Internet Key Exchange Protocol Version 2 (IKEv2), to negotiate IPsec ESP Bound End-to-End Tunnel (BEET) mode. BEET mode combines the benefits of tunnel mode with reduced overhead, making it suitable for applications requiring minimalistic end-to-end tunnels, mobility support, and multi-address multi-homing capabilities. The introduction of the USE_BEET_MODE Notify Message enables the negotiation and establishment of BEET mode security associations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 September 2025.

Copyright Notice

Table of Contents

1. Introduction

The Bound End-to-End Tunnel (BEET) mode, as specified in Appendix B of [RFC7402], offers an optimized approach for deploying IP Security (IPsec), [RFC4301], using Encapsulating Security Payload (ESP) [RFC4303] for end-to-end use cases. It combines the advantages of Tunnel and Transport modes specified in [RFC7296], while minimizing their overhead for end-to-end use cases.

The [RFC7402]does not specify necessary code points to negotiate a ESP BEET mode SA using the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296]. This document fills this gap by introducing a new Notify Message Status Type, USE_BEET_MODE, to facilitate the negotiation and establishment of BEET mode security associations in IKEv2.

1.1. Background

For over a decade, a minimalist IPsec tunnel mode, BEET, has been in use for end-to-end security in HIP environments without IKE negotiation, [RFC7401]. Also, in many environments, with IKE negotiation using a private IKEv2 Notify Message Status Type (strongSWAN).

Additionally, BEET mode ESP is valuable for low-power devices which usually use only one end-to-end IPsec tunnel, as it reduces power consumption [RFC9333]and complexity. In situations where devices or IPsec connections are dedicated to a single application or transport protocol, the use of BEET mode simplifies packet processing and conserves energy, especially benefiting lower-powered devices.

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

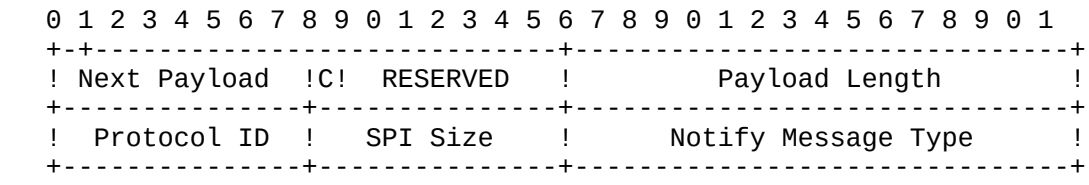
2. IKEv2 Negotiation

When negotiating a Child SA using IKEv2, the initiator MUST use the new "USE_BEET_MODE" Notify Message Status Type to request a Child SA pair with BEET mode support. The method used is similar to how USE_TRANSPORT_MODE is negotiated, as described in [RFC7296]

To request a BEET-mode SA on the Child SA pair, the initiator MUST include the USE_BEET_MODE, Notify Message Status Type, when requesting a new Child SA, either during the IKE_AUTH or the CREATE_CHILD_SA exchanges to create a new Child SA. If the request is accepted, the response MUST also include a USE_BEET_MODE Notification Message Status Type. If the responder declines and does not include the USE_BEET_MODE notification in the response, the child SA may be established without BEET mode enabled. If this is unacceptable to the initiator, the initiator MUST delete the child SA.

As the use of the USE_BEET_MODE mode payload is currently only defined for non-transport-mode tunnels, the USE_BEET_MODE notification MUST NOT be combined with the USE_TRANSPORT notification.

2.1. USE_BEET_MODE Notify Message Payload



* Payload Length - MUST be 0.

* Protocol ID (1 octet) - MUST be 0. MUST be ignored if not 0.

* SPI Size (1 octet) - MUST be 0. MUST be ignored if not 0.

3. IANA Considerations

This document defines a new "IKEv2 Notify Message Status Type" to be added to the IANA registry [STATUSNOTIFY]

Value	Notify Message Status Type	Reference
[TBD1]	USE_BEET_MODE	[this document]

4. Security Considerations

In this section we discuss the security properties of the BEET mode, discussing some and point out some of its limitations [RFC3552].

There are no known new vulnerabilities that the addition of the BEET mode to IKEv2 would create.

Since the BEET security associations have the semantics of a fixed, point-to-point tunnel between two IP addresses, it is possible to place one or both of the tunnel end points into other network or nodes but those that actually "possess" the inner IP addresses, i.e., to implement a BEET mode proxy. However, since such usage defeats the security benefits of combined ESP processing, as discussed in [I-D.nikander-esp-beet-mode], the implementations SHOULD NOT support such usage when used in combination with IKEv2; instead use IKEv2 MOBIKE to move the between networks.

5. Implementation Status

[Note to RFC Editor: Please remove this section and the reference to [RFC6982]before publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942].

The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Authors are requested to add a note to the RFC Editor at the top of this section, advising the Editor to remove the entire section before publication, as well as the reference to [RFC7942].

5.1. Linux XFRM

Linux

Organization: Linux kernel Project

Name: Linux Kernel https://www.kernel.org/

Description: Implements BEET mode in ESP. The initial support was added in 2006. It is widely used

Level of maturity: Stable and used for over 15 years

Licensing: GPLv2

Implementation experience: There is no support for IPv4 fragments yet. IPv6 fragments appears to work. The BEET mode code is in production for over a decade. And it appears stable.

Contact: https://lore.kernel.org/netdev/

5.2. strongSwan

Organization: The strongSwan Project

Name: strongSwan https://docs.strongswan.org/docs/5.9/swanctl/swanctlConf.html

Description: Implements IKE negotiation and ESP support for BEET mode Linux

Level of maturity: Stable for a long time

Coverage: Implements negotiating BEET mode support in Child SA negotiations and using it in ESP. The initial support was added in 2006.

Licensing: GPLv2

Implementation experience strongSwan use a private Notify Message Status Type USE_BEET_MODE (40961) for IKE. As far we know BEET is widely used.

Contact Tobias Brunner tobias@strongswan.org

5.3. iproute2

Organization: The iproute2 Project

Name: iproute2 https://git.kernel.org/pub/scm/network/iproute2/iproute2.git

Description: Implements BEET mode support in ESP. e.g. command support "ip xfrm policy ... mode beet" . and "ip xfrm state .. mode beet". The initial support was added in 2006

Level of maturity: Stable

Licensing: GPLv2

Implementation experience: TBD

Contact: https://lore.kernel.org/netdev/ or Stephen Hemminger stephen@networkplumber.org

6. Acknowledgment

We extend our sincere gratitude to the authors and contributors who contributed to the standardization of BEET mode. Their insights and dedication have significantly influenced our work, as well as their contributions to the implementation of BEET mode many years ago.

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7402] Jokela, P., Moskowitz, R., and J. Melen, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 7402, DOI 10.17487/RFC7402, April 2015, <https://www.rfc-editor.org/info/rfc7402>.

8. Informative References

[I-D.nikander-esp-beet-mode] Nikander, P. and J. Melen, "A Bound End-to-End Tunnel (BEET) mode for ESP", Work in Progress, Internet-Draft, draft-nikander-esp-beet-mode-09, 5 August 2008, <https://datatracker.ietf.org/doc/html/draft-nikander-esp-beet-mode-09>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013, <https://www.rfc-editor.org/info/rfc6982>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[RFC9333] Migault, D. and T. Guggemos, "Minimal IP Encapsulating Security Payload (ESP)", RFC 9333, DOI 10.17487/RFC9333, January 2023, <https://www.rfc-editor.org/info/rfc9333>.

[STATUSNOTIFY]
IANA, "IKEv2 Notify Message Status Types",
<https://www.iana.org/assignments/ikev2-parameters/
ikev2-parameters.xhtml#ikev2-parameters-16>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Antony Antony
secunet Security Networks AG
Email: antony.antony@secunet.com

Steffen Klassert
secunet Security Networks AG
Email: steffen.klassert@secunet.com