IPSECME Working Group
Internet-Draft
Updates: 4303 (if approved)
Intended status: Standards Track
Expires: 1 November 2025

L. Colitti
J. Linkova
Google
M. Richardson
Sandelman Software Works

30 April 2025

ESP Echo Protocol
draft-ietf-ipsecme-esp-ping-00

Abstract

This document defines an ESP echo function which can be used to detect whether a given network path supports ESP packets.

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 1 November 2025.

Table of Contents

1. Problem Statement 2
2. Terminology 3
3. Protocol Specification 3
4. Use cases 4
5. Discovering ESP Echo Support 5
6. Updates to RFC4303 5
7. Security Considerations 6
8. IANA Considerations 7
9. Acknowledgements 7
10. Changelog 7
11. References 7
11.1. Normative References 7
11.2. Informative References 7
Authors' Addresses 8

1. Problem Statement

IPsec sessions between nodes that have global connectivity will by default use ESP packets in IPv4 or IPv6 headers without encapsulation. These packets may have advantages over ESP-in-UDP encapsulation, such as:

* They require fewer keepalive packets to keep sessions open.

  ** On some networks, ESP is be statelessly allowed in both directions, and thus not require any keepalive packets at all. For example, the IPv6 Simple Security recommendations [RFC6092] specify that ESP by default must always be allowed and not be subject to any timeouts.

  ** Even if ESP is not statelessly allowed, experience from real world networks is that timeouts for ESP are higher than for UDP sessions, thus requiring IPsec endpoints to send fewer keepalives.

* They provide slightly lower overhead, due to the absence of the UDP header.

However, because ESP packets do not share fate with IKE packets, it is possible for the network to allow IKE packets but not ESP packets. This leads to the IPsec session not being able to exchange any packets even though IKE negotiation succeeded.

Because ESP is only used after IKE negotiation, this failure mode is difficult to predict, difficult to detect, and difficult to recover from. In particular, migrating a session using MOBIKE [RFC4555] to a network that does not allow ESP could result in the session blackholing all future packets until the problem is detected and a new migration is performed to enable encapsulation.

Operational experience suggests that networks and some home routers that drop ESP packets are common enough to be a problem for general purpose VPN applications desiring to work reliably on the Internet.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Protocol Specification

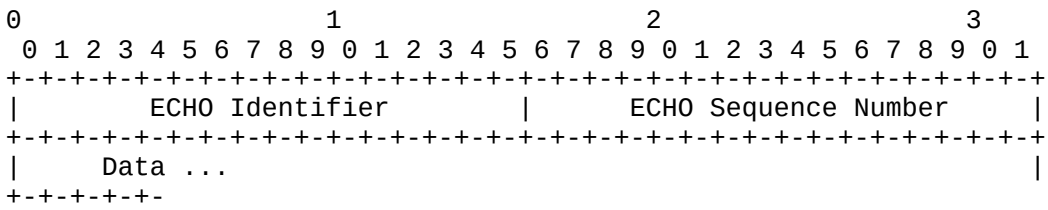
An IPv6 node that desires to determine whether the path to a particular destination can support ESP packets can send an ESP Echo Request packet to that destination. ESP Echo Request packets are ESP packets with an SPI value of (7-TBD) and a Next Header value of 59 (No Next Header).

If the destination supports ESP, and wishes to reveal to the sender that it does so, it SHOULD reply with an ESP Echo Reply packet. ESP Echo Reply packets are ESP packets with an SPI value of (8-TBD) and a Next Header value of 59.

The ESP Echo Request and Reply packets utilize the standard ESP packet format as described in Section 2 of [RFC4303] with the following changes:

* SPI set to
  - [ESP-ECHO-REQUEST] for ESP Echo Request
  - [ESP-ECHO-REPLY] for ESP Echo Reply
* The Next Header field of the ESP header SHOULD be set to 59 (No Next Header).
* No Integrity Check Value-ICV.

The payload has the following format:



* ECHO Identifier: An identifier to aid in matching Echo Replies to Echo Requests. MAY be zero.
  Implementations that support multiple simultaneous Echo Request sessions MUST ensure that different sessions have different identifiers. Implementations that are not aware of other implementations that might be running on the same node at the same time SHOULD randomize the identifier to prevent collisions, and MUST be prepared to receive responses to packets that were sent by another implementation.

* ECHO Sequence Number: An identifier to aid in matching Echo Replies to Echo Requests. MAY be zero.

* Data: Zero or more octets of arbitrary data.

Figure 1: ESP Echo Request and Reply Payload Overview

An IPsec peer, prior to an IKE negotiation or after completing an IPsec negotiation, intending to ascertain the path's capability to support ESP packets to a specific destination, MAY send one or more ESP Echo Request packet(s) to the destination. Should the destination support ESP and intend to communicate this capability to the potential IPsec peer, it SHOULD respond with an ESP Echo Reply packet.

The sender MAY send ESP Echo packets with zero data. When responding to an ESP Echo packet, the node MUST copy the data from the ESP Echo packet to the ESP Echo Reply packet, up to the limit of the MTU of the path back to the sender.

4. Use cases

A node that wishes to set up an IPsec session to a peer that is known to support this protocol can discover whether the intermediate network will carry ESP packets by sending an ESP Echo Request to the peer. Depending on wether it receives an ESP Echo Reply or not, it could choose to enable encapsulatior, use a different IP protocol, or use a different server or interface. For example, if MOBIKE [RFC4555] is used, a node can use ESP Echo Request packets to verify reachability before moving to a new address.

Network operators can troubleshoot IPsec sessions by sending ESP Echo Request packets from one peer to another to determine if the network between the peers will successfully carry ESP, and if so, what maximum packet size the network is able to support.

ESP Echo Requests can be used as keepalives, to maintain firewall state entries if the network statefully filters ESP between endpoints.

5. Discovering ESP Echo Support

If no response is received to an ESP Echo Request packet, it can be caused by one of the following:

* the peer doesn’t support ESP Echo protocol.

* there is no end-to-end ESP connectivity.

* intermediate nodes allow regular ESP packets, but drop ESP packets that have SPIs in the reserved SPI range.

Without some prior knowledge about ESP Echo support by the remote side, the sender can not distibguish those two scenarios. Therefore the sender SHOULD NOT treat lack of response as an indicator of end-to-end connectivity issues until an explicit confirmation of ESP Echo support by the peer is received. Because ESP might still work even if intermediate nodes drop ESP Echo Request or ESP Echo Reply packets, senders SHOULD still attempt to use ESP if no alternative paths or protocols (e.g., UDP encapsulation) are available. The sender MAY use any means of obtaining the information about ESP Echo support, such as an explicit out-of-band configuration (for example, a VPN client might be configured to always use ESP Echo when communicating to the given VPN server).

6. Updates to RFC4303

Section 2.6 of [RFC4303] discusses "dummy" ESP packets, which are distinguishable by the Next Header value set to 59. As per [RFC4303] a receiver MUST be prepared to silently discard "dummy" packets. This document updates Section 2.6 of [RFC4303] to allow packets with the Next Header value of 59 to be processed, if SPI is set to [ESP-ECHO-REQUEST] or [ESP-ECHO-REPLY].

OLD TEXT:

A transmitter MUST be capable of generating dummy packets marked with this value in the next protocol field, and a receiver MUST be prepared to discard such packets, without indicating an error.

NEW TEXT:

A transmitter MUST be capable of generating dummy packets marked with this value in the next protocol field, and a receiver MUST be prepared to discard such packets, without indicating an error. A transmitter MUST NOT use the reserved SPI values [ESP-ECHO-REQUEST] or [ESP-ECHO-REPLY] for dummy packets. A receiver SHOULD NOT discard packets with the Next Header value set of 59, if those packets use the reserved SPI values. Packets with the reserved SPI values [ESP-ECHO-REQUEST] or [ESP-ECHO-REPLY] and the Next Header value set of 59 SHOULD be processed by the receiver as described in draft-colitti-ipsecme-esp-ping.

7. Security Considerations

If an IPsec sender uses ESP Echo Request packets to determine whether the path supports ESP, an intermediate node may drop ESP Echo packets to make the sender believe that the path does not support ESP even though it does. To prevent such downgrade attacks, IPsec nodes MUST NOT fall back to unencrypted mode of communication in case of ESP Echo failure. The node MAY switch to another path (e.g. via another interface) or another protocol (e.g. IPv4).

Intermediate nodes can can forge ESP Echo Reply packets to cause the sender to believe that the network supports ESP even though it doesn’t. This may result in ESP packets being blackholed and ESP sessions being unable to transmit or receive data. Intermediate nodes can achieve the same effect by allowing ESP packets with an SPI of 7 or 8, but dropping packets with any other SPI value. This failure mode already exists today, because intermediate networks can always choose to drop ESP packets.

The security considerations are similar to other unconnected request-reply protocols such as ICMPv6 echo. In particular:

* By sending an ESP Echo Request from a spoofed source address, an attacker could cause a server to send an ESP Echo Reply to that address. This does not constitute an amplification attack because the ESP Echo Reply is the same size as the ESP Echo Request. This can be prevented by implementing ingress filtering per BCP 38 [RFC2827].

* An attacker can use ESP Echo Request packets to determine whether a particular destination address is an ESP endpoint. This is not a new attack because any endpoint that supports ESP must also reply to IKE INIT packets.

8. IANA Considerations

This memo requests that IANA allocate two new values from the "Security Parameters Index (SPI)" registry. The following entry should be appended:

Number	Description	Reference
7-ESP-ECHO-REQUEST	ESP Echo Request	THIS DOCUMENT
8-ESP-ECHO-REPLY	ESP Echo Reply	THIS DOCUMENT

Table 1

9. Acknowledgements

Thanks to Tero Kivinen, Steffen Klassert, Andrew McGregor, Valery Smyslov and Paul Wouters for helpful discussion and suggestions.

10. Changelog

11. References

11.1. 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>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <https://www.rfc-editor.org/info/rfc6092>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11.2. Informative References

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <https://www.rfc-editor.org/info/rfc4555>.

Authors' Addresses

Lorenzo Colitti
Google
Email: lorenzo@google.com

Jen Linkova
Google
Email: furry13@gmail.com

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca