Network
Internet-Draft
Intended status: Standards Track
Expires: June 9, 2016

T. Pauly
Apple Inc.
S. Touati
Ericsson

December 7, 2015

TCP Encapsulation of IKEv2 and IPSec Packets
draft-pauly-ipsecme-tcp-encaps-02

Abstract

This document describes a method to transport IKEv2 and IPSec packets over a TCP connection for traversing network middleboxes that may block IKEv2 negotiation over UDP. This method, referred to as TCP encapsulation, involves sending all packets for tunnel establishment as well as tunneled packets over a TCP connection.

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 http://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 June 9, 2016.

Table of Contents

1. Introduction

IKEv2 [RFC7296] is a protocol for establishing IPSec tunnels, using IKE messages over UDP for control traffic, and using ESP messages (or ESP over UDP) for its data traffic. Many network middleboxes that filter traffic on public hotspots block all UDP traffic, including IKEv2 and IPSec, but allow TCP connections through since they appear to be web traffic. Devices on these networks that need to use IPSec (to access private enterprise networks, to route voice-over-IP calls to carrier networks, or because of security policies) are unable to establish IPSec tunnels. This document defines a method for encapsulating both the IKEv2 control messages as well as the IPSec data messages within a TCP connection.

Using TCP as a transport for IPSec packets adds a third option to the list of traditional IPSec transports:

1. Direct. Currently, IKEv2 negotiations begin over UDP port 500. If no NAT is detected between the initiator and the receiver, then subsequent IKEv2 packets are sent over UDP port 500 and IPSec data packets are sent using ESP [RFC4303].

2. UDP Encapsulation [RFC3948]. If a NAT is detected between the initiator and the receiver, then subsequent IKEv2 packets are sent over UDP port 4500 with four bytes of zero at the start of the UDP payload and ESP packets are sent out over UDP port 4500. Some peers default to using UDP encapsulation even when no NAT are detected on the path as some middleboxes do not support IP protocols other than TCP and UDP.

3. TCP Encapsulation. If both of the other two methods are not available or appropriate, both IKEv2 negotiation packets as well as ESP packets can be sent over a single TCP connection to the peer. This connection can itself use TLS [RFC5246] or other methods if needed. If the connection uses TLS, it will also be capable of traversing a web proxy [RFC2817].

1.1. Prior Work and Motivation

Encapsulating IKEv2 connections within TCP or TLS streams is a common approach to solve the problem of UDP packets being blocked by network middleboxes. The goal of this document is to promote interoperability by providing a standard method of framing IKEv2 and ESP message within streams, and to provide guidelines for how to configure and use TCP encapsulation.

Some previous solutions include:

Cellular Network Access Interworking Wireless LAN (IWLAN) uses IKEv2 to create secure connections to cellular carrier networks for making voice calls and accessing other network services over Wi-Fi networks. 3GPP has recommended that IKEv2 and ESP packets be sent within a TLS connection to be able to establish

connections on restrictive networks.

ISAKMP over TCP Various non-standard extensions to ISAKMP have been deployed that send IPSec traffic over TCP or TCP-like packets.

SSL VPNs Many proprietary VPN solutions use a combination of TLS and IPSec in order to provide reliability.

IKEv2 over TCP IKEv2 over TCP as described in [I-D.nir-ipsecme-ike-tcp] is used to avoid UDP fragmentation.

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 RFC 2119 [RFC2119].

2. Configuration

One of the main reasons to use TCP encapsulation is that UDP traffic may be entirely blocked on a network. Because of this, support for TCP encapsulation is not specifically negotiated in the IKEv2 exchange. Instead, support for TCP encapsulation must be pre-configured on both the initiator and the responder.

The configuration defined on each peer should include the following parameters:

- One or more TCP ports on which the responder will listen for incoming connections. Note that the initiator may initiate TCP connections to the responder from any local port.

- Whether or not to use TLS for connections to a given TCP port. The responder may expect to read encapsulated IKEv2 and ESP packets directly from the TCP connection, or it may expect to read them from a stream of TLS data packets. The initiator should be pre-configured to use TLS or not when communicating with a given port on the responder.

Since TCP encapsulation of IKEv2 and IPSec packets adds overhead and

has potential performance trade-offs compared to direct or UDP-encapsulated tunnels (as described in Performance Considerations, Section 7), implementations SHOULD prefer IKEv2 negotiation over UDP.

3. TCP-Encapsulated Header Formats

In order to encapsulate IKEv2 and ESP messages within a stream (which may be raw TCP, or TLS over TCP), a 32-bit length field precedes every message. If the first 32-bits of the message are zeros (a Non-ESP Marker), then the contents comprise an IKEv2 message. Otherwise, the contents comprise an ESP message.

Although a TCP stream may be able to send very long messages, implementations SHOULD limit message lengths to typical UDP datagram ESP payload lengths. The maximum message length is used as the effective MTU for connections that are being encrypted using ESP, so the maximum message length will influence characteristics of inner connections, such as the TCP Maximum Segment Size (MSS).

3.1. TCP-Encapsulated IKEv2 Header Format

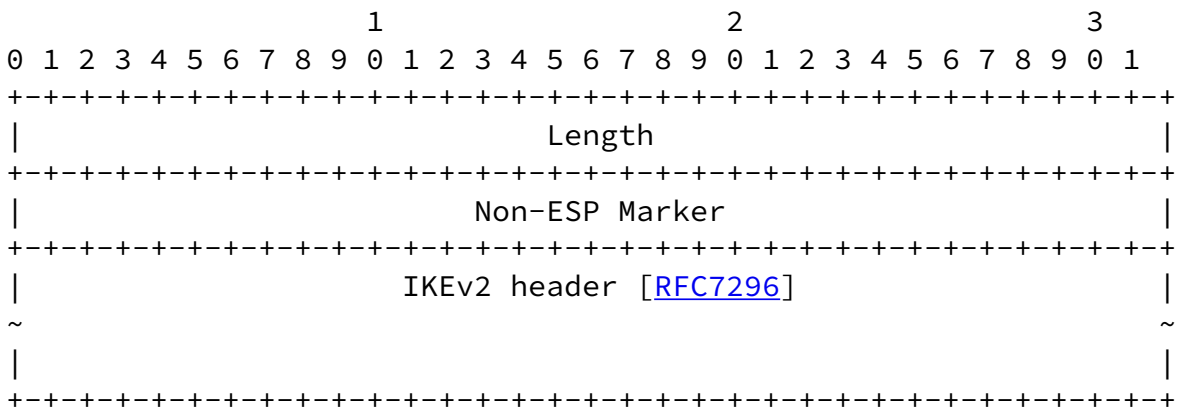

Figure 1

The IKE header is preceded by a 32-bit length field in network byte order that specifies the length of the IKE packet within the TCP stream. As with IKEv2 over UDP port 4500, a zeroed 32-bit Non-ESP Marker is inserted before the start of the IKEv2 header in order to differentiate the traffic from ESP traffic between the same addresses and ports.

- Length (4 octets, unsigned integer) - Length of the IKE packet including the Length Field and Non-ESP Marker.

3.2. TCP-Encapsulated ESP Header Format

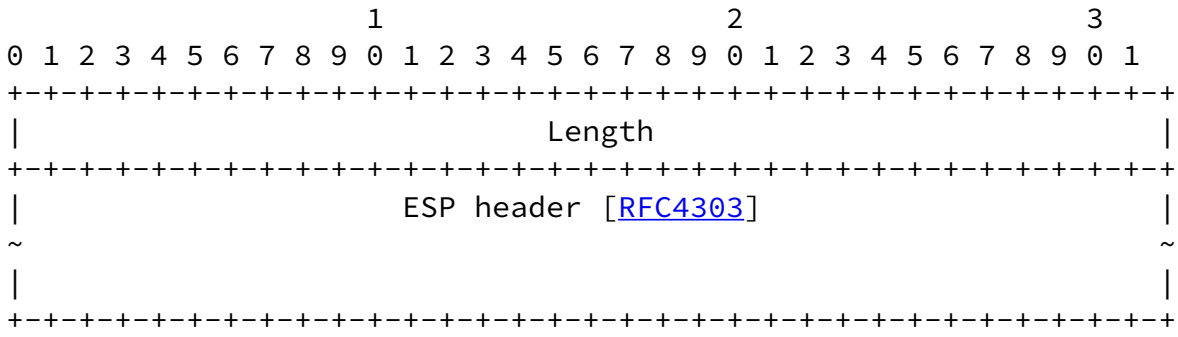

Figure 2

The ESP header is preceded by a 32-bit length field in network byte order that specifies the length of the ESP packet within the TCP stream.

- Length (4 octets, unsigned integer) - Length of the ESP packet including the Length Field.

4. Applicability

TCP encapsulation is applicable only when it has been configured to be used with specific IKEv2 peers. If a responder is configured to use TCP encapsulation, it MUST listen on the configured port(s) in case any peers will initiate new IKEv2 sessions. Initiators MAY use TCP encapsulation for any IKEv2 session to a peer that is configured to support TCP encapsulation, although it is recommended that initiators should only use TCP encapsulation when traffic over UDP is blocked.

Any specific IKE SA, along with its Child SAs, is either TCP encapsulated or not. A mix of TCP and UDP encapsulation for a single SA is not allowed. The exception to this rule is SAs that are migrated between addresses using MOBIKE (Section 7).

5. Connection Establishment and Teardown

When the initiator decides to use TCP encapsulation for IKEv2 negotiation, the initiator will initiate a TCP connection with the responder using the configured TCP port. If TLS is being used, it may be negotiated at this point, although the policy for the TLS negotiation is out of scope of this document. If a web proxy is applied to the ports for the TCP connection, and TLS is being used, the initiator can send an HTTP CONNECT message to establish a tunnel through the proxy [RFC2817]

Before either initiator or responder closes the TCP connection by sending a FIN or a RST, session teardown SHOULD be gracefully negotiated with DELETE payloads. Once all SAs have been deleted, the initiator of the original connection MUST close the TCP connection.

An unexpected FIN or a RST on the TCP connection may indicate either a loss of connectivity, an attack, or some other error. If a DELETE payload has not been sent, both sides SHOULD maintain the state for their SAs for the standard lifetime or time-out period. The original initiator (that is, the endpoint that initiated the TCP connection and sent the first IKE_SA_INIT message) is responsible for re-establishing the TCP connection if it is torn down for any unexpected reason. Since new TCP connections may use different ports due to NAT mappings or local port allocations changing, the responder MUST allow packets for existing SAs to be received from new source ports.

A peer must discard a partially received message due to a broken connection.

The streams of data sent over any TCP connection used for this protocol MUST begin with a complete IKEv2 message, complying to the

format specified in Figure 1. The stream must start with an IKE message to allow the peer to know with which IKE session the traffic is associated. If the session had been fully established previously, it is suggested to send an UPDATE_SA_ADDRESSES message when using MOBIKE, or an INFORMATIONAL message (a keepalive) when not using MOBIKE. If either initiator or responder receives a stream that cannot be parsed correctly, it MUST close the TCP connection.

Multiple TCP connections between the initiator and the responder are allowed, but their use must take into account the initiator capabilities and the deployment model: connecting to multiple gateways handling different ESP SAs deployed in a high availability model, for example. IKE and IPSec messages MUST be processed according to the standard source identification (using the SPI) and ordering rules independantly of whether a single or multiple TCP connections are used. It is possible to negotiate multiple IKE SAs over the same TCP connection, in which case messages are de-multiplexed using the SPI of the message.

6. Interaction with NAT Detection Payloads

When negotiating over UDP port 500, IKE_SA_INIT packets include NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP payloads to determine if UDP encapsulation of IPSec packets should be used. These payloads contain SHA-1 digests of the SPIs, IP addresses, and ports. IKE_SA_INIT packets sent on a TCP connection SHOULD include these payloads, and SHOULD use the applicable TCP ports when creating and checking the SHA-1 digests.

If a NAT is detected due to the SHA-1 digests not matching the expected values, no change should be made for encapsulation of subsequent IKEv2 or ESP packets, since TCP encapsulation inherently supports NAT traversal. Implementations MAY use the information that a NAT is present to influence keep-alive timer values.

7. Using MOBIKE with TCP encapsulation

When an IKEv2 session is transitioned between networks using MOBIKE [RFC4555], the initiator of the transition may switch between using TCP encapsulation, UDP encapsulation, or no encapsulation. Implementations that implement both MOBIKE and TCP encapsulation MUST support dynamically enabling and disabling TCP encapsulation as interfaces change.

The encapsulation method of ESP packets MUST always match the encapsulation method of the IKEv2 negotiation, which may be different when an IKEv2 endpoint changes networks. When a MOBIKE-enabled initiator changes networks, the UPDATE_SA_ADDRESSES notification

SHOULD be sent out first over UDP before attempting over TCP. If

there is a response to the UPDATE_SA_ADDRESSES notification sent over UDP, then the ESP packets should be sent directly over IP or over UDP port 4500 (depending on if a NAT was detected), regardless of if a connection on a previous network was using TCP encapsulation. Similarly, if the responder only responds to the UPDATE_SA_ADDRESSES notification over TCP, then the ESP packets should be sent over the TCP connection, regardless of if a connection on a previous network did not use TCP encapsulation.

8. Considerations for Keep-alives and DPD

Encapsulating IKE and IPSec inside of a TCP connection can impact the strategy that implementations use to detect peer liveness and to maintain middlebox mappings. In addition to mechanisms in IKE and IPSec, TCP keepalives are available. The following mechanisms may be employed:

- IKEv2 Informational packets [RFC7296]
- IPSec ESP NAT keep-alives [RFC3948]
- TCP NAT keep-alives [RFC1122]
- TLS keep-alives [RFC6520]

It is up to the implementation to decide which keepalives are appropriate for TCP-encapsulated connections. NAT timeouts are generally longer for TCP ports, but implementations should still use some form of keep-alive when a NAT is detected. If TCP NAT keep-alives are used, IPSec ESP NAT keep-alives may be considered redundant and can safely be disabled.

9. Middlebox Considerations

Many security networking devices such as Firewalls or Intrusion Prevention Systems, network optimization/acceleration devices and Network Address Translation (NAT) devices keep the state of sessions that transverse through them.

These devices commonly track the transport layer and/or the application layer data to drop traffic that is anomalous or malicious in nature.

A network device that monitors up to the application layer will commonly expect to see HTTP traffic within a TCP socket running over port 80, if non-HTTP traffic is seen (such as IKEv2 within TCP), this could be dropped by the security device. In this scenario the IKEv2

and ESP packets are to be TLS encapsulated (using port 443) to ensure that the security device permits the traffic.

In the case of a TLS proxy, where the network security device decrypts, inspects and re-encrypts the session the same considerations must be taken into account as for HTTP, where TLS can be inspected. In this case the TLS proxy must allow TCP encapsulation of IKEv2 and IPsec otherwise the session could be dropped.

A network device that monitors the transport layer will track the state of TCP sessions, it is common that the security device will perform actions such as tracking TCP sequence numbers.

10. Performance Considerations

Several aspects of TCP encapsulation for IKEv2 and IPSec packets may negatively impact the performance of connections within the tunnel. Implementations should be aware of these and take these into consideration when determining when to use TCP encapsulation.

10.1. TCP-in-TCP

If the outer connection between IKEv2 peers is over TCP, inner TCP connections may suffer effects from using TCP within TCP. In particular, the inner TCP's round-trip-time estimation will be affected by the burstiness of the outer TCP. This will make loss-recovery of the inner TCP traffic less reactive and more prone to spurious retransmission timeouts.

10.2. Added Reliability for Unreliable Protocols

Since ESP is an unreliable protocol, transmitting ESP packets over a TCP connection will change the fundamental behavior of the packets. Some application-level protocols that prefer packet loss to delay (such as Voice over IP or other real-time protocols) may be negatively impacted if their packets are retransmitted by the TCP connection due to packet loss.

10.3. Encryption Overhead

If TLS or another encryption method is used on the TCP connection, there may be increased processing overhead for encrypting and decrypting. This overhead may be experienced as a decrease in throughput on CPU-limited devices, or an increase in CPU usage or battery consumption on other devices, therefore the initiator and

responder MUST allow the selection of NULL cipher when using TLS.

Additionally, the TLS record introduces another layer of overhead, requiring more bytes to transmit a given IKEv2 and IPSec packet.

11. Security Considerations

IKEv2 responders that support TCP encapsulation may become vulnerable to new Denial-of-Service (DoS) attacks that are specific to TCP, such as SYN-flooding attacks. Responders should be aware of this additional attack-surface.

Attackers may be able to disrupt the TCP connection by sending spurious RST packets. Due to this, implementations SHOULD make sure that IKE session state persists even if the underlying TCP connection is torn down.

If TLS is used on the encapsulating TCP connection it should not be considered as a security measure, and no TLS profile is recommended by this specification. The security of the IKEv2 session is entirely derived from the IKEv2 negotiation and key establishment.

12. IANA Considerations

This memo includes no request to IANA.

TCP port 4500 is already allocated to IPSec. This port MAY be used for the protocol described in this document, but implementations MAY prefer to use other ports based on local policy. We foresee some implementations using TCP port 443 to more easily pass through some middleboxes [I-D.tschofenig-hourglass].

13. Acknowledgments

The authors would like to acknowledge the input and advice of Stuart Cheshire, Delziel Fernandes, Yoav Nir, Christoph Paasch, Yaron Sheffer, David Schinazi, March Wu and Kingwel Xie. Special thanks to Eric Kinnear for his implementation work.

14. References

14.1. 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, <http://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, <http://www.rfc-editor.org/info/rfc7296>.

14.2. Informative References

[I-D.nir-ipsecme-ike-tcp] Nir, Y., "A TCP transport for the Internet Key Exchange", draft-nir-ipsecme-ike-tcp-01 (work in progress), July 2012.

[I-D.tschofenig-hourglass] Tschofenig, H., "The New Waist of the Hourglass", draft-tschofenig-hourglass-00 (work in progress), July 2012.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[RFC2817] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, DOI 10.17487/RFC2817, May 2000, <http://www.rfc-editor.org/info/rfc2817>.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, DOI 10.17487/RFC3948, January 2005, <http://www.rfc-editor.org/info/rfc3948>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005,

<http://www.rfc-editor.org/info/rfc4303>.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <http://www.rfc-editor.org/info/rfc6520>.

Authors' Addresses

Tommy Pauly
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
US

Email: tpauly@apple.com

Samy Touati
Ericsson
300 Holger Way
San Jose, California 95134
US

Email: samy.touati@ericsson.com