Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2012

Y. Nir
Check Point

June 13, 2012

A TCP transport for the Internet Key Exchange
draft-nir-ipsecme-ike-tcp-00

Abstract

This document describes using TCP for IKE messages. This facilitates the transport of large messages over paths where fragments are dropped.

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 December 15, 2012.

1. Introduction

The Internet Key Exchange (IKE) specified in [RFC2407] and [RFC2408], and IKEv2 as specified in [RFC5996] uses UDP to transport the exchange messages. Some of those messages may be fairly large. Specifically, the 5th and 6th messages of IKEv1 Main Mode, the first and second messages of IKEv1 Aggressive Mode, and the messages of IKEv2 IKE_AUTH exchange can become quite large, as they may contain a chain of certificates, a signature payload (called "Auth" in IKEv2), CRLs, and in the case of IKEv2, some configuration information that is carried in the CFG payload.

When such UDP packets exceed the path MTU, they get fragmented. This increases the probability of packets getting dropped, but the retransmission mechanisms in IKE (as described in section 2.1 of RFC 5996) takes care of that. More recently we have seen a number of service providers dropping fragmented packets. Firewalls and NAT devices need to keep state for each packet where some but not all of the fragments have been received. This creates a burden in terms of memory, especially for high capacity devices such as Carrier-Grade NAT (CGN) or high capacity firewalls.

The BEHAVE working group has an Internet Draft describing required behavior of CGNs ([CGN-reqs]). It requires CGNs to comply with [RFC4787], which in section 11 requires NAT devices to support fragments. However, some people deploying IKE have found that some ISPs have begun to drop fragments in preparation for deploying CGNs. While we all hope for a future where all devices comply with the emerging standards, and where CGNs are not required, we have to make IKE work today.

The solution described in this document is to transport the IKE messages over a TCP ([RFC793]) rather than over UDP. IKE packets (both versions) describe their own length, so they are well-suited for transport over a stream-based connection such as TCP. The Initiator opens a TCP connection to the Responder's port 500, sends the requests and receives the responses, and then closes the connection. TCP can handle arbitrary-length messages, works well with any sized data, and is well supported by all ISP infrastructure.

1.1. Non-Goals of this Specification

Firewall traversal is not a goal of this specification. If a firewall has a policy to block IKE and/or IPsec, hiding the IKE exchange in TCP is not expected to help. Some implementations hide both IKE and IPsec in a TCP connection, usually pretending to be HTTPS by using port 443. This has a significant impact on bandwidth and gateway capacity, and even this is defeated by better firewalls.

SSL VPNs tunnel IP packets over TLS, but the latest firewalls are also TLS proxies, and are able to defeat this as well.

This document is not part of that arms race. It is only meant to allow IKE to work When faced with broken infrastructure that drops large IP packets.

1.2. Conventions Used in This Document

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. The Protocol

2.1. Initiator

An Initiator MAY try IKE using TCP. It opens a TCP connection from an arbitrary port to port 500 of the Responder. When the three-way handshake completes, the Initiator MUST send the request. If the Initiator knows that this request is the last request needed at this time, it SHOULD half-close the TCP connection, although it MAY wait until the last response is received. When all responses have been received, the Initiator MUST close the connection. If the peer has closed the connection before all requests have been transmitted or responded to, the Initiator SHOULD either open a new TCP connection or transmit them over UDP again.

It MUST accept responses sent over IKE within the same connection, but MUST also accept responses over other transports, if the request had been sent over them as well.

2.2. Responder

A Responder MAY accept TCP connections to port 500, and if it does, it MUST accept IKE requests over this connection. Responses to requests received over this connection MUST also go over this connection. If the connection has closed before the Responder had had a chance to respond, it MUST NOT respond over UDP, but MUST instead wait for a retransmission over UDP or over another TCP connection.

The responder MUST accept different requests on different transports. Specifically, the Responder MUST NOT rely on subsequent requests coming over the same transport. For example, it is entirely acceptable to have the first two requests on IKE Main Mode come over UDP port 500, while the last request comes over TCP, and the

following Quick Mode request might come over UDP port 4500 (because NAT has been detected).

If the responder has some requests of its own to send, it MUST NOT use a connection that has been opened by a peer. Instead, it MUST either use UDP or else open a new TCP connection to the original Initiator's TCP port 500.

The normal flow of things is that the Initiator opens a connection and closes its side first. The responder closes after sending the last response where the initiator has already half-closed the connection. If, however, a significant amount of time has passed, and neither new requests arrive nor the connection is closed by the initiator, the Responder MAY close or even reset the connection.

This specification makes no recommendation as to how long such a timeout should be, but a few seconds should be enough.

2.3. Transmitter

The transmitter, whether an initiator transmitting a request or a responder transmitting a response MUST NOT retransmit over the same connection. TCP takes care of that. It SHOULD send the IKE header and the IKE payloads with a single command or in rapid succession.

2.4. Receiver

The IKE header is copied from RFC 5996 below for reference:

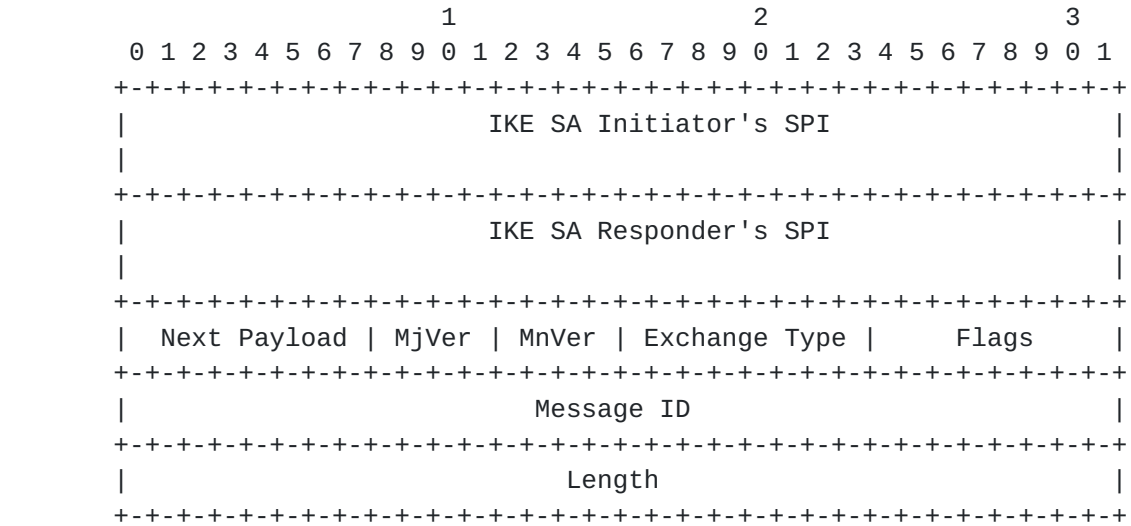

Figure 1: IKE Header Format

The receiver MUST first read in the 28 bytes that make up the IKE

header. The Responder then subtracts 28 from the length field, and reads the resulting number of bytes. The combined message, comprised on 28 header bytes and whatever number of payload bytes is processed the same way as regular UDP messages. That includes retransmission detection, with one slight difference: if a retransmitted request is detected, the response is retransmitted as well, but using the current TCP connection rather than whatever other transport had been used for the original transmission of the request.

3. Operational Considerations

Most IKE is relatively short messages. Quick Mode in IKEv1, and in IKEv2 all but the IKE_AUTH exchange are short. It is only the IKE_AUTH exchange in IKEv2. UDP has advantages in lower latency and lower resource consumption, so it makes sense to use UDP whenever TCP is not required.

The requirements in Section 2.2 mean that different requests may be sent over different transports. So the initiator can choose the transport on a per-request basis. So one obvious policy would be to do everything over UDP except the specific requests that tend to become too big. This way the first messages use UDP, and the Initiator can set up the TCP connection at the same time, eliminating the latency penalty of using TCP. This may not always be the most efficient policy, though. It means that the first messages sent over TCP are relatively large ones, and the way TCP works means that client (or initiator) will wait for an ACK before transmitting the second segment of the IKE request.

An alternative method, that is probably easier for the Initiator to implement, is to do an entire "mission" using the same transport. So if TCP is needed and an IKE SA has not yet been created, the Initiator will open a TCP connection, and perform all 2-4 requests needed to set up a child SA over the same connection.

Yet another policy would be to begin by using UDP, and at the same time set up the TCP connection. If at any point the TCP handshake completes, the next requests go over that connection. This method can be used to auto-discover support of TCP on the responder. This is easier for the user than configuring which peers support TCP, but has the potential of wasting resources, as TCP connections may finish the three-way handshake just when IKE over UDP has finished. The requirements from the responder ensure that all these policies will work.

4. Security Considerations

Most of the security considerations for IKE over TCP are the same as those for UDP as in RFC 5996.

For the Responder, listening to TCP port 500 involves all the risks of maintaining any TCP server. Precautions against DoS attacks, such as SYN cookies are RECOMMENDED.

5. IANA Considerations

No IANA action is required for this specification, as TCP port 500 is already allocated to "ISAKMP".

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2408, November 1998.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol: IKEv2", RFC 5996, September 2010.

6.2. Informative References

[CGN-reqs] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements (work in progress), May 2012.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", RFC 4787, January 2007.

[RFC793] Postel, J., "Transmission Control Protocol", RFC 793, STD 7, September 1981.

Author's Address

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim st.
Tel Aviv 67897
Israel

Email: ynir@checkpoint.com