IPsecME Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 14, 2014

Y. Nir
Check Point

August 13, 2013

Adopting Child SAs Following Re-Authentication in IKEv2
draft-nir-ipsecme-cafr-00

Abstract

This document describes an extension to the IKEv2 protocol whereby Child SAs are moved to the new IKE SA following re-authentication. This allows for a smoother transition with no loss of connectivity.

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 February 14, 2014.

Table of Contents

1. Introduction

The Internet Key Exchange version 2 (IKEv2) protocol, as specified in [RFC5996bis] associates Child SAs with the IKE SAs under which the exchange that created them took place. With the deletion of the IKE SA due to expiry, policy change, or an explicit message from the peer, the child SAs associated with it are implicitly closed as described in section 1.4.1 of the IKEv2 document. This behavior is not desired when IKE SAs are replaced rather than deleted, because those child SAs could still be valid and there is no security reason to create new ones prematurely.

There are two cases where an IKE SA is replaced.

1. Rekeying, where new keys are generated. This is described in section 2.18 of RFC 5996. This is done mainly for key freshness.
2. Re-Authentication, where both sides authenticate, and new keys are generated. This is done as part of a risk management policy, to limit the time that compromised IKE SA keys can be used to provide the attacker access to the network. No reauthentication exchange is specified in the RFC. Instead, it's simply the Initial and Authentication exchanges done as if from scratch. This is described in section 2.8.3 of RFC 5996.

For rekeying, RFC 5996 provides a way to avoid having to re-create all child SAs. When an IKE SA is rekeyed, all the Child SAs under the old IKE SA are inherited by the new IKE SA, so that the subsequent deletion of the old IKE SA does not affect the Child SAs. This behavior is described in section 2.8 paragraph 4 of RFC 5996.

For reauthentication, RFC 5996 does not provide a similar mechanism, and section 2.8.3 explicitly says that Child SAs need to be created from scratch. This is often inconvenient, as IPsec systems usually create Child SAs only in response to traffic and multiple Child SAs may exist for a single IKE SA. The protocol extension in this draft closes this gap.

1.1. 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].

The terms IKE SA, Child SA, Rekeying, and Reauthentication are as described in the RFC 5996.

2. Adopting Child SAs

This document defines a new notification that is added to the IKE_AUTH exchange that is used to re-authenticate. The notification proves that the current participant in the IKE_AUTH exchange is the same one that had participated in the old IKE SA. If both peers send this notification, and it verifies correctly, all Child SAs belonging to the old IKE SA are immediately inherited by the new IKE SA.

In addition to the Child SAs, any IP address assigned to either peer through the use of the CFG payload (as described in section 2.19 of RFC 5996, is also associated with the new IKE SA.

Following a successful re-authentication exchange, the old IKE SA is deleted by the Initiator.

2.1. The ADOPT_CHILD_SAS Notification

The ADOPT_CHILD_SA notification is formatted as follows:

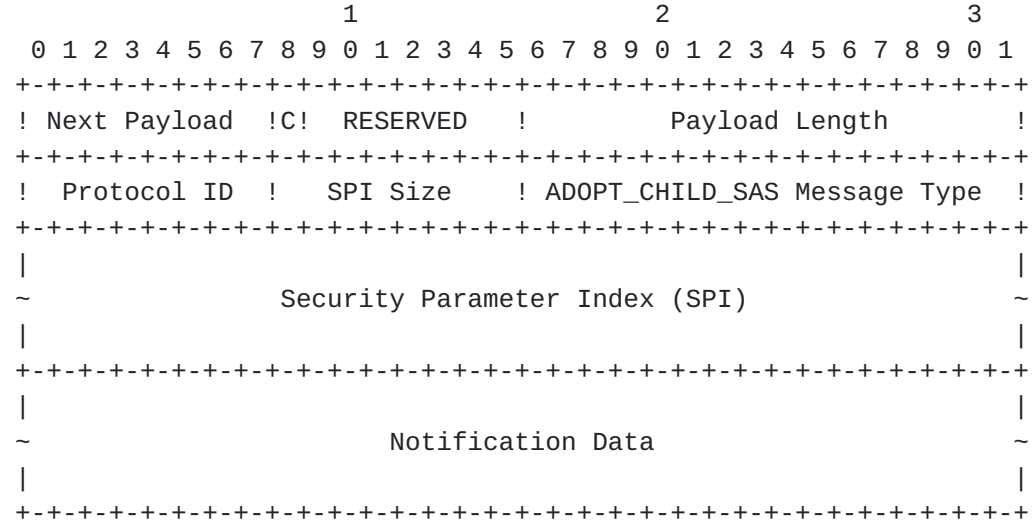

Figure 1

- Protocol ID (1 octet) MUST be 1, denoting an IKE SA. Note that previous versions of RFC 5996 explicitly mentioned the possibility, but the current version omits this as prior to this specification there were no cases where the value 1 should have been used.
- SPI Size (1 octet) MUST be 16, as that is the size of the concatenation of the IKE SPIs.
- Security Parameter Index (16 octets) - contains the concatenated SPIs of the old IKE SA. The Initiator SPI comes first, similar to the first 16 bytes of the IKE header.

- ADOPT_CHILD_SAS Notify Message Type (2 octets) - MUST be xxxxx, the value assigned for ADOPT_CHILD_SAS. TBA by IANA.
- Notification Data (variable) - contains the proof of ownership of the previous IKE SA. Calculation of this field is described in Section 2.2.

2.2. Calculating the Proof of Possession Value

The notification data field in the ADOPT_CHILD_SAS notification is calculated as follows:

```
InitiatorPOP = prf(SK_pi, "Adopting Child SAs for Initiator")

ResponderPOP = ptr(SK_pr, "Adopting Child SAs for Responder")
```

InitiatorPOP and ResponderPOP are respectively sent the initiator and responder in the IKE_AUTH exchange that creates the reauthenticated IKE SA. The roles may be reversed from those of the original IKE SA, but it is still the new Initiator that uses the old SK_pi value. The algorithms used, the PRF keys and the length of the output are all those from the old IKE SA, not the new one.

2.3. Verifying the Proof of Possesion Value

Both sides of the IKE_AUTH exchange should be in possession of the SK_pi and SK_pr values from the previous IKE SA. This allows both sides to make the calculation and verify that it is correct. This verification MUST be done only after the other side has been authenticated. If the value does not verify, the IKE_AUTH exchange MUST be terminated, and an INVALID_SYNTAX notification MUST be sent.

To go through with the new IKE SA inheriting the SAs of the old IKE SA, all of the following MUST apply:

- Both sides have to be successfully authenticated.
- The authenticated identities of both sides are the same as those in the old SA. If the authenticated identity of one peer differs from the authenticated identity that it had in the previous IKE SA, the other side MUST respond with an INVALID_SYNTAX notification. See Section 3 for a discussion of a possible race condition.
- The proof of possession values in the ADOPT_CHILD_SAS notification both validated. The responder MUST NOT continue in sending the last IKE_AUTH packet if this condition is not satisfied. See Section 3 for a discussion of what happens if the responder's notification does not validate.

3. Dealing With the Possible Race Condition

The sections above describe two kinds of failures in the IKE_AUTH exchange:

1. An authentication failure. This could be something as sinister as an attack, or as innocent as a temporary failure to contact an OCSP server.
2. A validation failure of the ADOPT_CHILD_SAS notification.

If either of those failures occurs for the Initiator, there is no problem. The IKE_AUTH exchange is aborted, the old IKE SA is still valid, and all the Child SAs belong to that old IKE SA.

If, however, the failure occurs for the Responder, we may have a problem. Having sent the last IKE_AUTH response, the responder is confident that the exchange has completed successfully, and can transfer the Child SAs to the new IKE SA. However, when the Initiator sees that last response, one of the two errors happens, and this leads it to delete the new IKE SA. The Responder erases the new IKE SA, deleting with it all the Child SAs. The result is a mismatch in databases, where the Initiator still has the valid SAs, while the Responder does not.

If the Child SAs have been transferred, and the new IKE SA has been deleted, but the old IKE SA has not yet been deleted, then the Responder MUST delete the old IKE SA (using a DELETE payload) immediately after receiving the deletion of the new IKE SA. If the Child SAs have not yet been transferred, then the Responder MAY keep the old IKE SA along with the Child SAs until they are deleted by the peer or expire according to policy.

The Initiator MUST NOT delete the old IKE SA because of a failure of IKE to create a new IKE SA. The old IKE SA may only be deleted if policy dictates it, such as when a reauthentication timer expires.

Following a successful verification and transfer of the Child SAs, the Initiator SHOULD delete the old IKE SA.

4. Interaction with Other Standards

This document changes things so that there is often no need to create new Child SAs along with the new IKE SA when reauthenticating. This makes the full IKE_AUTH exchange with the piggy-backed Child SA exchange (as described in RFC 5996) superfluous. Implementations should consider implementing the childless extension of IKEv2 ([RFC6023] in addition to this specification.

5. IANA Considerations

IANA is requested to assign a notify message type from the status types range (16418-40959) of the "IKEv2 Notify Message Types" registry with name "ADOPT_CHILD_SAS"

6. Security Considerations

Comparing the authenticated identities of the new IKE SA with those of the old IKE SA is critical. Without it, attackers would be able to authenticate as themselves, steal the Child SAs, and then close them. The proof of possession seems to be superfluous, and in most cases it really is. However, there are some uses of IKE by multiple entities with a shared identity and a shared credential. Calculating and verifying the proof of possession blocks such entities from stealing each others SAs.

An on-path attacker may get the Initiator to send the ADOPT_CHILD_SAS notification before failing authentication. This notification is a PRF calculated with a secret key over a known message. The security properties of PRFs are such that this does not reveal any secret data such as IKE SA keys.

7. Changes from Previous Versions

First version

8. References

8.1. Normative References

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

[RFC5996bis] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", draft-kivinen-ipsecme-ikev2-rfc5996bis-00 (work in progress), August 2013.

8.2. Informative References

[RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)", RFC 6023,

October 2010.

Author's Address

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim st.
Tel Aviv 6789735
Israel

Email: ynir@checkpoint.com