Using Isakmp Message Ids for Replay Protection

Status of this Memo

This document is a submission to the IETF Internet Protocol Security (IPsec) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@lists.tislabs.com) or to the editor.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete 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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

This document is a product of the IETF’s IPsec Working Group. Copyright (C) The Internet Society (2001). All Rights Reserved.
Abstract

This document describes a method for adding explicit replay protection to IKE messages by altering the use of the message id in the ISAKMP header.

We suggest that this change could be applied as part of the next revision of IKE.
# Table of Contents

1. Introduction ................................................... 4  
2. Specification of Requirements .................................. 4  
3. Outline of the Problem ........................................ 4  
4. Discussion of Proposed Solutions .............................. 5  
5. Description of the new behaviour ................................ 6  
6. Cryptographic Impact ............................................ 7  
7. IANA Considerations ............................................. 7  
8. Security Considerations ........................................ 7  
9. References .................................................... 7
1. Introduction

IKE has a number of small weaknesses that should be fixed in the next version. Many of these weaknesses can be avoided through careful software design, but they still exist as pitfalls for the unwary implementer. An unfortunate side effect of these small weaknesses is that different implementers have chosen to solve them in different ways.

For example, the Initial Contact notification is not authenticated by the phase 1 hash. Thus, some implementations will only send it in encrypted phase 1 messages; others may always send it attached to a quick mode or in a separate informational exchange.

Fortunately, many of these weaknesses are quite easy to fix. For example, the problem of unauthenticated payloads is fixed simply by redefining the phase 1 hash. In this memo, we discuss another weakness of IKE which can be fixed with a fairly simple fix: the problem of replay protection.

2. Specification of Requirements

This document shall use the keywords "MUST", "MUST NOT", "REQUIRED","SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" to describe requirements. They are to be interpreted as described in [Bradner].

3. Outline of the Problem

The ISAKMP message format does not provide explicit protection against replay attacks. This does not necessarily mean that IKE is vulnerable to replay attacks, but it does complicate the overall protocol design because implementers must be vigilant in ensuring that each individual IKE exchange is replay-resistant.

As described in [PROP], it takes a minimum of 3 messages before an IKE exchange can be secured against replay. This is precisely the reason why quick mode is 3 messages long when 2 messages would otherwise appear to suffice.

An implementation that performs any time-consuming or memory-consuming operation before the 3rd message has been exchanged does so at its own risk. This caveat can often conflict with operational needs, such as the desire for optimization.
For example, if PFS is used in quick mode, the responder may be tempted to precompute the Diffie-Hellman secret before the exchange is complete in order to set up the SA as quickly as possible. This would introduce a security hole, because it allows an attacker to replay the first packet of quick mode, causing DoS.

Replay of quick mode packets can be detected because the attacker will be unable to forge the 3rd packet of the exchange. Replayed info mode packets cannot be detected in this way because they are not part of a 3-message exchange. However, replayed info modes will have different potential weaknesses depending on the content of the message. A replayed delete notification is unlikely to cause any harm because the SA will have already been deleted.

On the other hand, if an implementation tried to avoid the aforementioned phase 1 hash weakness by putting the Initial Contact notification in a separate info mode message, an attacker could capture this packet and replay it later, possibly causing the peer to delete all his SAs.

The replay vulnerability also applies to proposed extensions to IKE. For example, two separate proposals for adding a heartbeat protocol to IKE each had to devote a section of the protocol design to thwarting replay attacks.

4. Discussion of Proposed Solutions

There is a minor related problem, which is that the 32-bit number space for message ids is not adequately collision resistant. It is possible, although fairly unlikely, that two peers may simultaneously initiate new exchanges using the same message id. If the messages crossed on the wire, each side would interpret the newly received message as a reply to the packet that it just sent.

Two categories of solutions have been proposed to fix these problems:

1. Turn the message id into an anti-replay counter.
2. Store all message ids that have been sent or received.

We believe that option 1 is a simpler and more elegant solution for several reasons. Most notably, it preserves the property already present in IKE that an IKE SA has a fixed memory footprint which does not increase over time.

This property is of theoretical importance because it guarantees that a properly designed implementation cannot be forced into an unstable
state under low memory conditions. In most applications, the memory
requirement for storing all these message ids is quite small.
However, some implementations may consume message ids more quickly
than this (for example, a host which negotiates multiple selector-
bound SAs or which sends frequent keep-alive packets).

Under option 2, an implementation may be forced to rekey its phase 1
SAs simply to free up some memory. The worse the memory problem gets,
the more frequently the host will need to rekey. This could cause an
implementation to enter an unstable (looping) state.

Solution 1 also prevents a message from being intercepted, stored,
and then sent at a later time. Why risk this potential vulnerability
when a simple, more elegant solution exists?

5. Description of the new behaviour

Instead of choosing a pseudorandom message id, the message id can be
converted into a counter. As in all counter-based anti-replay
schemes, each side should keep a record of the last message id it
received from the peer. (Obviously, this field should only be updated
after the packet has been authenticated)

The initiator (of the phase 1) uses 1 as his first message id, and he
increments this value by 1 on each subsequent exchange. The responder
(from the phase 1) does the same, but he always sets the high bit of
his message id to 1. This ensures that the initiator and responder
can never generate the same value.

Just as for the replay counters in \[ARCH\], an implementation should
keep a sliding window of which replay counters are acceptable. This
takes into account the possibility that some packets will be received
out of order.

We suggest that these packet-processing rules be incorporated into
the next version of IKE. They may also be enabled in the short term
by mutual exchange of the vendor id 0x325df29a2319f2dd.

This vendor id SHOULD NOT be used unless the two peers can
authenticate that the behaviour was agreed on (e.g. by the use of the
revised hash), due to the DoS attack that could occur if an attacker
caused only one peer to think that the new behaviour was being used.
6. Cryptographic Impact

This draft redefines the use of the message id field. The message id is used as an input to a hash function in order to generate the initial IV for an exchange. This IV is completely based on publicly visible material and can be generated by a passive snoop.

The only consequence of this change is that subsequent inputs to the hash will all have very low Hamming distance. Assuming that a good hash function is used, this shouldn’t affect the apparent randomness of the IV.

If a preponderance of cryptographers believed that this reduction of entropy was unacceptable, then the IV derivation could be changed in order to increase the entropy in the input to the PRF.

7. IANA Considerations

This document does not require any assigned numbers.

8. Security Considerations

The focus of this document is security; hence security considerations permeate this specification.

9. References


[iKE] Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998

Authors’ Addresses

Andrew Krywaniuk  
Alcatel Networks Corporation  
600 March Road  
Kanata, ON  
Canada, K2K 2E6  
+1 (613) 784-4237  
E-mail: andrew.krywaniuk@alcatel.com

The IPsec working group can be contacted via the IPsec working group’s mailing list (ipsec@lists.tislabs.com) or through its chairs:

Theodore Y. Ts’o  
tyts@MIT.EDU  
Massachusetts Institute of Technology

Barbara Fraser  
byfraser@cisco.com  
Cisco Systems

Expiration

This document expires February 9th, 2002.