Network Working Group
Request for Comments: 3104
Category: Experimental

G. Montenegro
Sun Microsystems, Inc.
M. Borella
CommWorks
October 2001

RSIP Support for End-to-end IPsec

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

The IESG notes that the set of documents describing the RSIP technology imply significant host and gateway changes for a complete implementation. In addition, the floating of port numbers can cause problems for some applications, preventing an RSIP-enabled host from interoperating transparently with existing applications in some cases (e.g., IPsec). Finally, there may be significant operational complexities associated with using RSIP. Some of these and other complications are outlined in section 6 of the RFC 3102, as well as in the Appendices of RFC 3104. Accordingly, the costs and benefits of using RSIP should be carefully weighed against other means of relieving address shortage.

Abstract

This document proposes mechanisms that enable Realm Specific IP (RSIP) to handle end-to-end IPsec (IP Security).

Table of Contents

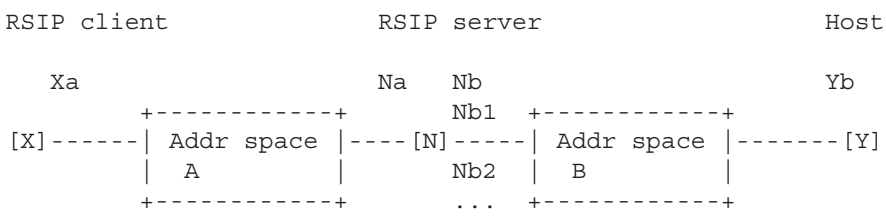
1. Introduction

This document specifies RSIP extensions to enable end-to-end IPsec. It assumes the RSIP framework as presented in [RSIP-FW], and specifies extensions to the RSIP protocol defined in [RSIP-P]. Other terminology follows [NAT-TERMS].

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.

2. Model

For clarity, the discussion below assumes this model:



Hosts X and Y belong to different address spaces A and B, respectively, and N is an RSIP server. N has two addresses: Na on address space A, and Nb on address space B. For example, A could be a private address space, and B the public address space of the general Internet. Additionally, N may have a pool of addresses in address space B which it can assign to or lend to X.

This document proposes RSIP extensions and mechanisms to enable an RSIP client X to initiate IKE and IPsec sessions to a legacy IKE and IPsec node Y. In order to do so, X exchanges RSIP protocol messages with the RSIP server N. This document does not yet address IKE/IPsec session initiation from Y to an RSIP client X. For some thoughts on this matter see Appendix G.

The discussion below assumes that the RSIP server N is examining a packet sent by Y, destined for X. This implies that "source" refers to Y and "destination" refers to Y's peer, namely, X's presence at N.

This document assumes the use of the RSAP-IP flavor of RSIP (except that port number assignments are optional), on top of which SPI values are used for demultiplexing. Because of this, more than one RSIP client may share the same global IP address.

3. Implementation Notes

The RSIP server N is not required to have more than one address on address space B. RSIP allows X (and any other hosts on address space A) to reuse Nb. Because of this, Y's SPD SHOULD NOT be configured to support address-based keying. Address-based keying implies that only one RSIP client may, at any given point in time, use address Nb when exchanging IPsec packets with Y. Instead, Y's SPD SHOULD be configured to support session-oriented keying, or user-oriented keying [Kent98c]. In addition to user-oriented keying, other types of identifications within the IKE Identification Payload are equally effective at disambiguating who is the real client behind the single address Nb [Piper98].

Because it cannot rely on address-based keying, RSIP support for IPsec is similar to the application of IPsec for remote access using dynamically assigned addresses. Both cases impose additional requirements which are not met by minimally compliant IPsec implementations [Gupta]:

> Note that a minimally-compliant IKE implementation (which only implements Main mode with Pre-shared keys for Phase I authentication) cannot be used on a remote host with a dynamically assigned address. The IKE responder (gateway) needs to look up the initiator's (mobile node's) pre-shared key before it can

decrypt the latter's third main mode message (fifth overall in Phase I). Since the initiator's identity is contained in the encrypted message, only its IP address is available for lookup and must be predictable. Other options, such as Main mode with digital signatures/RSA encryption and Aggressive mode, can accommodate IKE peers with dynamically assigned addresses.

IKE packets are typically carried on UDP port 500 for both source and destination, although the use of ephemeral source ports is not precluded [ISAKMP]. IKE implementations for use with RSIP SHOULD employ ephemeral ports, and should handle them as follows [IPSEC-MSG]:

> IKE implementations MUST support UDP port 500 for both source and destination, but other port numbers are also allowed. If an implementation allows other-than-port-500 for IKE, it sets the value of the port numbers as reported in the ID payload to 0 (meaning "any port"), instead of 500. UDP port numbers (500 or not) are handled by the common "swap src/dst port and reply" method.

It is important to note that IPsec implementations MUST be aware of RSIP, at least in some peripheral sense, in order to receive assigned SPIs and perhaps other parameters from an RSIP client. Therefore, bump-in-the-stack (BITS) implementations of IPsec are not expected to work "out of the box" with RSIP.

4. IKE Handling and Demultiplexing

If an RSIP client requires the use of port 500 as its IKE source, this prevents that field being used for demultiplexing. Instead, the "Initiator Cookie" field in the IKE header fields must be used for this purpose. This field is appropriate as it is guaranteed to be present in every IKE exchange (Phase 1 and Phase 2), and is guaranteed to be in the clear (even if subsequent IKE payloads are encrypted). However, it is protected by the Hash payload in IKE [IKE]. Because of this, an RSIP client and server must agree upon a valid value for the Initiator Cookie.

Once X and N arrive at a mutually agreeable value for the Initiator Cookie, X uses it to create an IKE packet and tunnels it the RSIP server N. N decapsulates the IKE packet and sends it on address space B.

The minimum tuple negotiated via RSIP, and used for demultiplexing incoming IKE responses from Y at the RSIP server N, is:

- IKE destination port number
- Initiator Cookie
- Destination IP address

One problem still remains: how does Y know that it is supposed to send packets to X via Nb? Y is not RSIP-aware, but it is definitely IKE-aware. Y sees IKE packets coming from address Nb. To prevent Y from mistakenly deriving the identity of its IKE peer based on the source address of the packets (Nb), X MUST exchange client identifiers with Y:

- IDii, IDir if in Phase 1, and
- IDci, IDcr if in Phase 2.

The proper use of identifiers allows the clear separation between those identities and the source IP address of the packets.

5. IPsec Handling and Demultiplexing

The RSIP client X and server N must arrive at an SPI value to denote the incoming IPsec security association from Y to X. Once N and X make sure that the SPI is unique within both of their SPI spaces, X communicates its value to Y as part of the IPsec security association establishment process, namely, Quick Mode in IKE [IKE] or manual assignment.

This ensures that Y sends IPsec packets (protocols 51 and 50 for AH and ESP, respectively) [Kent98a,Kent98b] to X via address Nb using the negotiated SPI.

IPsec packets from Y destined for X arrive at RSIP server N. They are demultiplexed based on the following minimum tuple of demultiplexing fields:

- protocol (50 or 51)
- SPI
- destination IP address

If N is able to find a matching mapping, it tunnels the packet to X according to the tunneling mode in effect. If N cannot find an appropriate mapping, it MUST discard the packet.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.