The ISAKMP Configuration Mode
<draft-ietf-ipsec-isakmp-mode-cfg-00.txt>

Status of this Memo

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

This document is an Internet-Draft. 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 draft documents are 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."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.

Abstract

This document describes a new ISAKMP mode that allows configuration items to be set by using both push/acknowledge and request/reply paradigms.
1. Introduction

ISAKMP provides a framework to negotiate and derive Security Associations. But since it is used within the IPSec framework, it may also be used to configure secure hosts. This configuration may take place between a gateway, an end-host client, or a configuration manager. For example, this can be used to configure multi-protocol IP tunnels securely.

It is assumed that the reader is familiar with the terms and concepts described in the "Security Architecture for the Internet Protocol" [Atkinson95] and "IP Security Document Roadmap" [Thayer97] documents.

Readers are advised to be familiar with both [Harkins97] and [ISAKMP] because of the terminology used within this document and the fact that this document is an extension of both of those documents.

1.1. Specification of Requirements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [Bradner97].
2. Configuration Mode

Configuration Mode uses an XCHG type of 34 for the ISAKMP header. This mode MUST NOT be used prior to establishment of an ISAKMP Phase 1 Security Association.

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR*, HASH, CFG</td>
<td>HDR*, HASH, CFG</td>
</tr>
</tbody>
</table>

where HASH is the prf output, using SKEYID_a as the key, and the message-ID from the ISAKMP header concatenated with the entire CFG payload, including attributes. In other words the hashes for the above exchange are:

\[ \text{HASH} = \text{prf}(\text{SKEYID}_a, \text{M-ID} | \text{CFG}) \]

Only one CFG payload MAY be present in one exchange of this mode.

3. Configuration Transaction

A "Configuration Transaction" is defined as two Configuration Mode exchanges, the first being either a Set or a Request and the second being either a Acknowledge or a Reply respectively. The Message ID within the ISAKMP header identifies the configuration transaction and MUST NOT be zero.

There are two paradigms to follow for this mode.

- "Set/Acknowledge" works on the push principle that allows a configuration manager (a host that wishes to send information to another) to start the configuration transaction. The Acknowledge code MUST return zero length attributes that it accepted. Those attributes that it did not accept will NOT be sent back in the acknowledgement.

- "Request/Reply" allows a host to request information from an informed host (a configuration manager). Attributes in the Request exchange may have values filled in to request these values once again. The Reply exchange MAY wish to choose those values, or return new values. It MAY also add new attributes and not include some requested attributes.

Transactions are completed once the Reply or Acknowledge code is received. If one is not received, the implementation MAY wish to retransmit the original exchange.
If a badly formatted exchange is received, the exchange SHOULD be dropped and logged locally, as per local policy. Badly formatted exchanges would also include those with unknown codes or unknown attributes within the Configuration Mode.

3.1. Configuration Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISKAMP_CFG_REQUEST</td>
<td>1</td>
</tr>
<tr>
<td>ISKAMP_CFG_REPLY</td>
<td>2</td>
</tr>
<tr>
<td>ISKAMP_CFG_SET</td>
<td>3</td>
</tr>
<tr>
<td>ISKAMP_CFG_ACK</td>
<td>4</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>5 - 127</td>
</tr>
<tr>
<td>Reserved for private use</td>
<td>128 - 255</td>
</tr>
</tbody>
</table>

4. Configuration Payload

The Configuration Payload is used to accomplish configuration transactions between two secure hosts. Its "next payload" value is 13.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload |   RESERVED    |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |   RESERVED    |     Number of Attributes      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Attributes                                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Configuration Payload fields are defined as follows:

- **Next Payload (1 octet)** - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0. For the Configuration Mode, this field MUST be 0, since there is never a next payload.

- **RESERVED (1 octet)** - Unused, set to 0.

- **Payload Length (2 octets)** - Length in octets of the entire Configuration payload, including the Configuration payload header and all of the attributes.
o Code (1 octet) - A code that represents a command to be fulfilled or an action that has been completed. Please see Section 4.1 for a description of each code.

o RESERVED (1 octet) - Unused, set to 0.

o Number of Attributes (2 octets) - States the number of attributes to follow. Note that both the payload length and the number of attributes field may be used to process the attributes.

o Attributes (variable) - One or more data attributes as defined in ISAKMP, section 3.3. Please see a later section for more information on the contents of these attributes.

5. Configuration Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERNAL_IP4_ADDRESS</td>
<td>1</td>
<td>Variable</td>
<td>4 octets</td>
</tr>
<tr>
<td>INTERNAL_IPX_ADDRESS</td>
<td>2</td>
<td>Variable</td>
<td>6 octets</td>
</tr>
<tr>
<td>INTERNAL_NB_ADDRESS</td>
<td>3</td>
<td>Variable</td>
<td>16 octets</td>
</tr>
<tr>
<td>INTERNAL_IP4_DNS</td>
<td>4</td>
<td>Variable</td>
<td>4 octets</td>
</tr>
<tr>
<td>INTERNAL_IP4_NBNS</td>
<td>5</td>
<td>Variable</td>
<td>4 octets</td>
</tr>
<tr>
<td>RENEW_SECONDS</td>
<td>6</td>
<td>Basic/Variable</td>
<td>2 or 4 octets</td>
</tr>
<tr>
<td>USE_IP4_DHCP</td>
<td>7</td>
<td>Variable</td>
<td>4 octets</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>8-63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved for private use</td>
<td>64-127</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

o INTERNAL_IP4_ADDRESS - Specifies an IPv4 address within the internal network. This address is sometimes called a red node address. This address is sometimes an illegal address on the Internet.

o INTERNAL_IPX_ADDRESS - Specifies an IPX address within the internal network.

o INTERNAL_NB_ADDRESS - Specifies a NetBios address within the internal network.

o INTERNAL_IP4_DNS - Specifies an IPv4 address of a DNS server within the network.

o INTERNAL_IP4_NBNS - Specifies an IPv4 address of a NetBios Name Server (WINS) within the network.
o RENEW_SECONDS - Specifies the number of seconds that the host can use all of the information set within the configuration transaction. The host MUST renew this information before this expiry time and MUST not use any of the information obtained through the configuration transaction after the expiry time.

o USE_IP4_DHCP - Instructs the host to request any subsequent information through the DHCP protocol. This attribute holds the IPv4 address of a DHCP server.

It is hoped that more attribute types will be defined in the future. Some suggestions would be to distribute local policy, or even authentication certificates. Also, note that no recommendations are made to how an implementation actually figures out what information to send. i.e. we do not recommend any specific method of (a gateway) determining which DNS server should be returned to a requesting host.

6. Security Considerations

This entire draft discusses a new ISAKMP mode to allow entities in the network to acquire and share configuration information.

The draft mandates that this exchange always occur after the Phase I Security Association has been set up and that the entire exchange be protected by the Phase I SA. Thus the exchange is as secure as any Phase II SA.

7. References


[Bradner97] Bradner, S., "Key words for use in RFCs to indicate Requirement Levels", RFC2119, March 1997


8. Acknowledgments

The editors would like to thank Peter Ford of Microsoft and Bob Moskowitz of Chrysler.

9. Editors’ Addresses

Roy Pereira
rpereira@timestep.com
TimeStep Corporation
+1 (613) 599-3610 x 4808

Sanjay Anand
sanjayan@microsoft.com
Microsoft Corporation
+1 (206) 936-6367

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

Robert Moskowitz
rgm@chrysler.com
Chrysler Corporation

Theodore Y. TsÂo
tytso@MIT.EDU
Massachusetts Institute of Technology