PPP Working Group
INTERNET DRAFT
Category: Informational
Title: draft-ietf-pppext-l2tp-sec-04.txt
Date: July 1998

Pat R. Calhoun
Sun Microsystems, Inc.
W. Mark Townsley
Cisco Systems
Sumit A. Vakil
VPNet Technologies, Inc.
Don Grosser
IBM Corporation

Layer Two Tunneling Protocol "L2TP" Security Extensions for Non-IP networks

<draft-ietf-pppext-l2tp-sec-04.txt>

Status of this Memo

Abstract

The L2TP document [1] defines the base protocol which describes the method of tunneling PPP [2] data. The L2TP document states that the security mechanism used over an IP network is to use the IETF's IPSEC protocols.

L2TP was designed in such a way as to be able to run over any underlying layer (i.e. Frame Relay, ATM, etc.). This document specifies extensions to the L2TP protocol in order to provide authentication and integrity of individual packets in a tunneled session over a network where IPSEC or another suitable security

protocol is not available.

Table of Contents

1.0 Introduction

The L2TP protocol specification states that the IPSEC protocols MUST be used over an IP network for L2TP to operate in a secure manner. However, L2TP may be run on a link layer that does not have a security mechanism such as IPSEC available. In this case it becomes necessary for L2TP to provide its own mechanism for packet level security.

This document will describe how authentication and integrity of L2TP packets will be handled over networks where IPSEC or another suitable security protocol does not exist. It does not intend to provide a mechanism for encryption of packets. If data encryption is necessary, then the user may utilize ECP or another form of end to end encryption.

The security extensions defined here also provide the added flexibility to negotiate security separately over the control and data channels. This may be desirable in some situations, particularly where processing power may be at a minimum, but some level of security is still desired.

By design, several of the constructs used here draw upon those being developed in the IPSEC working group.

1.1 Conventions

The following language conventions are used in the items of specification in this document:

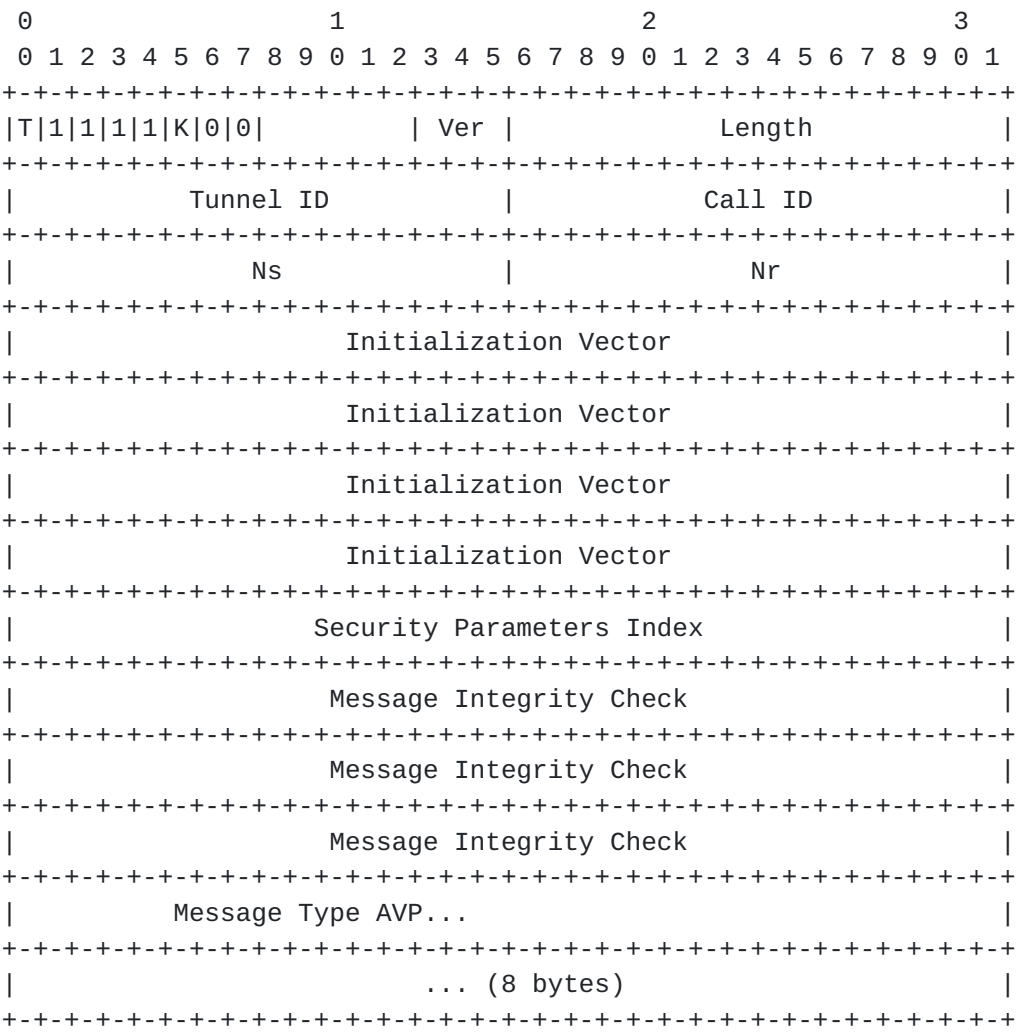
- MUST, SHALL, or MANDATORY -- This item is an absolute requirement of the specification.

- SHOULD or RECOMMEND -- This item should generally be followed for all but exceptional circumstances.

- MAY or OPTIONAL -- This item is truly optional and may be followed or ignored according to the needs of the implementor.

2.0 L2TP Security Header Format

The L2TP Header has been modified as follows in order to accommodate the new security extension.



K Bit

The 'K' bit MUST be set to one (1) when the security extension is present. The behavior of the 'K' bit is identical on both the control and data channel.

Initialization Vector (IV)

This field MUST contain a cryptographically random 128 bit value [5].

Security Parameters Index (SPI)

The SPI is an arbitrary four octet value. It is an unstructured

opaque index which is used in conjunction with the Tunnel ID to identify a particular Security Association. The behavior of the SPI is identical on the control and data channel.

The value inserted in this field is the locally generated SPI value which references the key used in generating the Message Integrity Check.

Message Integrity Check (MIC)

The MIC contains the result of the HMAC-MD5-96 algorithm [3][4] as applied over the entire L2TP packet. The behavior of the MIC is identical on the control and data channel.

3.0 Protection Against Attacks

This section will define certain methods of protecting against specific known types of attacks.

3.1 Denial of Service Attacks

There currently exists a Denial of Service Attack whereby a malicious host can issue a stream of Start-Control-Connection- Request messages to an L2TP host on a network.

Although an implementation MUST time-out when a Start-Control-Connection-Connected has not been received within a given window, there is still a possibility that if the messages were received fast enough the L2TP host would deplete its Control Connection Control Blocks. This form of attack is aggravated when the malicious host sends the packets with a random source IP address.

One form of protection against this attack is to have a local list of trusted hosts, however this does not scale very well when providing a roaming service from anywhere on the Internet. Furthermore, enforcing a security policy based on a source address is a very weak form of protection.

Another method of protecting against this form of attack is to have the 'K' bit set in the initial Start-Control-Connection-Request message. The message would be signed with the common secret (or key, see below for more details). This scheme will ensure that only authenticated Start-Control-Connection-Requests will be accepted, making this type of attack very inconvenient for a malicious user to create.

In order for this scheme to be successful, it is imperative that the base specification require that a base implementation which does not

support any extensions MUST reject a Start-Control-Connection-Request message with a 'K' bit set.

3.2 Replay Attacks

One common attack is the replay attack. This requires that a malicious user gain access to the network where packets are routed.

There are two different types of replay attacks in the current L2TP protocol. The first takes advantage of the fact that since a secret is a long lived key (known as the master key), a malicious user can retrieve the Stop-Control-Connection-Request message from two L2TP peers and replay it at a later date when an L2TP tunnel is active between both peers.

This form of attack is further complicated by the fact that the malicious user must inject the packet when the sequence number in the replayed packet is within the window of the receiver. This can be achieved using a brute force type attack by constantly sending the packet until the L2TP host accepts it. One more complication for the malicious user is the fact that the Tunnel and Call identifiers MUST be the same in the new session being attacked. This is possible, but improbable if the Tunnel and Call IDs are selected in a sufficiently random manner (while L2TP does not specify a method for selecting Tunnel and Call IDs, we reccommend choosing a method that is as unpredictable as possible to help guard against replay attacks, regardless if a security protocol is being utilized over the link).

The second type of attack occurs when a user attempts to replay data packets being tunneled. An example of a malicious packet to replay would be a LCP Terminate Request message from a previous session. In this case, again, the Tunnel and the Call IDs MUST be identical for the L2TP peer to accept the packet.

However, if a malicious user was to simply snoop the network and replay valid data packets from the current session it could potentially create some form of denial of service for the user. A good example of such a packet would be a TCP FIN packet (which are very common when using the WEB which have many short-lived connections). Since most TCP implementations do not have random initial sequence numbers, this is a very simple attack.

In order to protect against such an attack it is recommended that the L2TP flow control mechanism be enabled on the data path. This will offer protection since a replay packet would only be accepted once the window "rolled" over.

3.3 Compromise of the Master Key

Since tunnels may be long-lived and frequent, it is possible for the master key to be compromised. A malicious user could gain many valid samples and given enough resources could guess the master key. This is a very serious problem which must be addressed.

One simple and effective method to protect against this is to have both L2TP peers generate a session key when a tunnel is created. This key would be transmitted in the Start-Control-Connection-Request and the appropriate -Reply message. Furthermore, an L2TP peer could generate a new key whenever its sequence number "rolls" over. This would create a new security association between both peers, and protect against compromise of the master key.

This scheme would also protect against the replay of the data packet described above since the key would be changed once the Sequence number reached zero, making the replayed packet non- authenticated.

4.0 AVP Hiding

Document [1] states that a shared secret that exists between the tunnel peers is used for the AVP Hiding algorithm. However when using this extension on the control channel, the shared secret is only used to "hide" the initial control and payload channel key.

All subsequent AVP hiding uses the key instead of the shared (or master) key (including any other AVPs in the SCCRQ and SCCRP messages). This means that the key renegotiation procedure uses the old key to hide the new key.

5.0 Security Association Negotiation

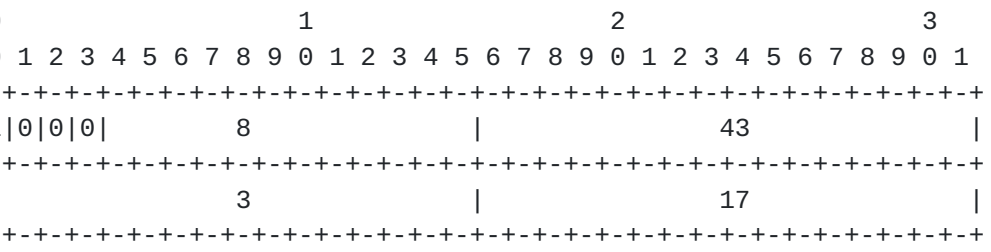
This section will define the new message type and AVPs which are required for the security extensions of the L2TP protocol. The AVPs allow designation of a Key for control messages, payload messages, or both. The Keys may or may not be the same for each.

5.1 Renegotiate-Security-Association (RSA)

The Renegotiate-Security-Association message type is a new L2TP control message used to renegotiate a new security association. It MAY be sent periodically while the control connection is established. To avoid certain replay attacks It SHOULD be sent before the sequence number of a control or call queue "rolls" back to 0.

If the CallId field in the L2TP header was set to a zero value, the key is being renegotiated for the control channel. If a non-zero value was found the key is being renegotiated for the payload channel.

Renegotiate-Security-Association



The Message Type AVP is encoded with a Vendor ID of 43 (3Com Corporation) with the attribute set to 3, mandatory, indicating Renegotiate-Security-Association. This AVP MUST be present. This message type indicates that the peer wishes to negotiate a new key for the payload stream or control stream. If the message contains a new key for the control channel the message digest function is calculated using the decrypted form of the key found within the Encoded Message Key AVP found in this message.

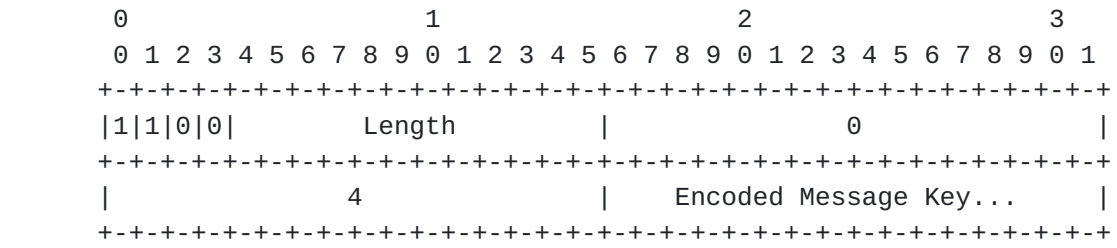
5.2 Encoded Message Key

The Encoded Message Key AVP may be present in SCCRQ, SCCRP, OCRQ, OCRP, ICRQ and ICRP. This message is used to inform the tunnel peer of a key to be used for the generation of the MIC (shown above) as well as the hiding of all attributes [1].

The presence of this attribute in the SCCRQ and SCCRP indicates that the key is being setup for the control channel. When found in the OCRQ, OCRP, ICRQ or the ICRP, the key is to be used for the payload channel which is referenced by the Assigned CallId AVP.

Note that there is a direct relationship between this key and the SPI value passed in the same message.

When a key is being negotiated on the control channel, any AVP hiding MUST use the decrypted form of this key as the shared secret [1].

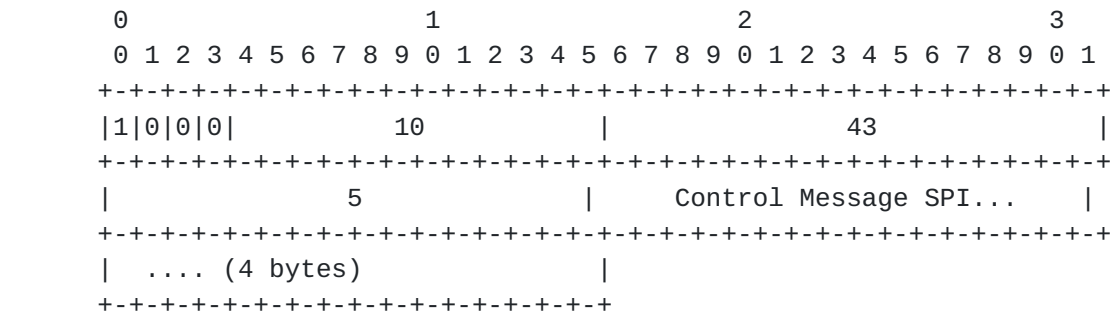


The Message Type AVP is encoded with a Vendor ID of 43 (3Com Corporation) with the attribute set to 4, mandatory, with the indicated number of bytes representing the encoded message key. This AVP MAY be present in the messages shown above. This AVP MUST be hidden and is optional. When present, the L2TP peer is indicating that authentication is required on all control or payload packets.

5.3 Message Security Parameter Index

The SPI is used as a reference to a session key. The locally generated SPI value MUST be inserted in the SPI field of the L2TP header to reference the appropriate KEY (found in the Encoded Message Key AVP).

When renegotiating a new key, a new SPI MUST be generated to correctly identify the key. The old key MUST become invalidated. On the payload channel, the peer MAY retain the old SPI for a short time in order to authenticate packets which are received out of order.



The Message Type AVP is encoded with a Vendor ID of 43 (3Com Corporation) with the attribute set to 5, mandatory, with length 10. This AVP MUST be present if the Encoded Message Key AVP is present. This AVP is optional and contains the security parameters index for the control or the payload channel.

6.0 Acknowledgments

We would like to thank Baiju Patel from Intel Coproration for their assistance.

7.0 References

[1] K. Hamzeh, T. Kolar, M. Littlewood, G. Singh Pall, J. Taarud, A. J. Valencia, W. Verthein, W.M. Townsley, B. Palter, A. Rubens "Layer Two Tunneling Protocol (L2TP)", Internet draft, October 1997

[2] W. Simpson, "The Point-to-Point Protocol (PPP)", RFC 1661, 07/21/1994

[3] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997

[4] R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, 04/16/1992

[5] D. Eastlake III, S. Crocker, J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994

8.0 Authors' Addresses

Questions about this memo can be directed to:

Pat R. Calhoun
Technology Development
Sun Microsystems, Inc.
15 Network Circle
Menlo Park, California, 94025
USA

Phone: 1-650-786-7733
Fax: 1-650-786-6445
E-mail: pcalhoun@eng.sun.com

W. Mark Townsley
Cisco Systems
7025 Kit Creek Road
P.O. Box 14987
Research Triangle Park, NC 27709
USA

Phone: 919-472-3741
Fax: 919-472-2940
E-mail: townsley@cisco.com

Sumit A. Vakil
VPNet Technologies, Inc.
1530 Meridian Ave
San Jose, CA 95125
USA

Phone: 408-445-6600 x264
Fax: 408-445-6611
E-mail: svakil@vpnet.com

Don Grosser
IBM Corporation
700 Park Office Dr.
Research Triangle Park, NC, 27709
USA

Phone: 919-254-2160
E-Mail: grosser@raleigh.ibm.com

Appendix A: Additional Recommendations for secure L2TP implementations

This appendix identifies some potential security problems with the L2TP and includes reccommendations for ways to avoid the associated risks. We do not identify any new protocol entities here, rather provide implementation advice for greater security when using L2TP.

A.1 Proxy CHAP

While proxy CHAP provides a useful method of forwarding the challenge issued by the LAC and the response from the client to the LNS for final processing, there is a potential security risk if the operator of the LNS does not FULLY trust the operator of the LAC. Granted, there must be some level of trust between these two entities to setup billing practices, etc. However, allowing the LAC to control the challenge gives the operator of the LAC a very simple (and perhaps tempting) way to impersonate any user which has been tunneled through her system in the past (given that the user's password has not changed in the home network). By simply replaying the challenge/response pair to the LNS in the proxy CHAP AVP, a malicious user can gain access as that user on the home network via the LNS at any time. This impersonated call can continue to exist undetected until a CHAP rechallenge is sent from the LNS to the client at which time the fake client will presumably fail to answer the challege

correctly and be disconnected.

Neither the protocol specified in this document nor IPSEC can counter against this kind of attack by a malicious, yet "trusted" LAC. However, the LNS can remedy this problem by simply issuing a CHAP rechallenge so that the challenge is issued by the LNS rather than the LAC. This makes it much more difficult for the LAC operator to spoof the CHAP authentication phase at your LNS, reducing vulnerabilty considerably.

To implement this security feature, a CHAP rechallenge MUST be issued from the LNS in lieu of sending a CHAP SUCCESS based upon the proxy CHAP values sent from the LAC. If the proxy CHAP values sent from the LAC result in a CHAP FAILURE, there is no compelling reason to send the rechallenge unless you wish to give the client another "chance" at answering the challenge correctly.

A.2 Tunnel ID and Call ID selection

As suggested in section 2.2, Tunnel IDs and Call IDs SHOULD be selected in a sufficiently random manner rather than sequentially or any other predictable order. Doing so helps prevent a malicious user who otherwise does not have access to packet traces to and from a LAC or LNS to guess the ID of an active session and attempt to hijack it.

However, when using the L2TP Security extensions this requirement is no longer required since the level of authentication this extension provides does not allow a malicious user to simply guess a Tunnel and Call Id.