Network Working Group
Request for Comments: 3193
Category: Standards Track

B. Patel
Intel
B. Aboba
W. Dixon
Microsoft
G. Zorn
S. Booth
Cisco Systems

November 2001

Securing L2TP using IPsec

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document discusses how L2TP (Layer Two Tunneling Protocol) may utilize IPsec to provide for tunnel authentication, privacy protection, integrity checking and replay protection. Both the voluntary and compulsory tunneling cases are discussed.

Table of Contents

1. Introduction

L2TP [1] is a protocol that tunnels PPP traffic over variety of networks (e.g., IP, SONET, ATM). Since the protocol encapsulates PPP, L2TP inherits PPP authentication, as well as the PPP Encryption Control Protocol (ECP) (described in [10]), and the Compression Control Protocol (CCP) (described in [9]). L2TP also includes support for tunnel authentication, which can be used to mutually authenticate the tunnel endpoints. However, L2TP does not define tunnel protection mechanisms.

IPsec is a protocol suite which is used to secure communication at the network layer between two peers. This protocol is comprised of IP Security Architecture document [6], IKE, described in [7], IPsec AH, described in [3] and IPsec ESP, described in [4]. IKE is the key management protocol while AH and ESP are used to protect IP traffic.

This document proposes use of the IPsec protocol suite for protecting L2TP traffic over IP networks, and discusses how IPsec and L2TP should be used together. This document does not attempt to

standardize end-to-end security. When end-to-end security is required, it is recommended that additional security mechanisms (such as IPsec or TLS [14]) be used inside the tunnel, in addition to L2TP tunnel security.

Although L2TP does not mandate the use of IP/UDP for its transport mechanism, the scope of this document is limited to L2TP over IP networks. The exact mechanisms for enabling security for non-IP networks must be addressed in appropriate standards for L2TP over specific non-IP networks.

1.1. Terminology

Voluntary Tunneling
: In voluntary tunneling, a tunnel is created by the user, typically via use of a tunneling client. As a result, the client will send L2TP packets to the NAS which will forward them on to the LNS. In voluntary tunneling, the NAS does not need to support L2TP, and the LAC resides on the same machine as the client. Another example of voluntary tunneling is the gateway to gateway scenario. In this case the tunnel is created by a network device, typically a router or network appliance. In this scenario either side may start the tunnel on demand.

Compulsory Tunneling
: In compulsory tunneling, a tunnel is created without any action from the client and without allowing the client any choice. As a result, the client will send PPP packets to the NAS/LAC, which will encapsulate them in L2TP and tunnel them to the LNS. In the compulsory tunneling case, the NAS/LAC must be L2TP-capable.

Initiator
: The initiator can be the LAC or the LNS and is the device which sends the SCCRQ and receives the SCCRP.

Responder
: The responder can be the LAC or the LNS and is the device which receives the SCCRQ and replies with a SCCRP.

1.2. Requirements language

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [2].

2. L2TP security requirements

L2TP tunnels PPP traffic over the IP and non-IP public networks. Therefore, both the control and data packets of L2TP protocol are vulnerable to attack. Examples of attacks include:

[1] An adversary may try to discover user identities by snooping data packets.

[2] An adversary may try to modify packets (both control and data).

[3] An adversary may try to hijack the L2TP tunnel or the PPP connection inside the tunnel.

[4] An adversary can launch denial of service attacks by terminating PPP connections, or L2TP tunnels.

[5] An adversary may attempt to disrupt the PPP ECP negotiation in order to weaken or remove confidentiality protection. Alternatively, an adversary may wish to disrupt the PPP LCP authentication negotiation so as to weaken the PPP authentication process or gain access to user passwords.

To address these threats, the L2TP security protocol MUST be able to provide authentication, integrity and replay protection for control packets. In addition, it SHOULD be able to protect confidentiality for control packets. It MUST be able to provide integrity and replay protection of data packets, and MAY be able to protect confidentiality of data packets. An L2TP security protocol MUST also provide a scalable approach to key management.

The L2TP protocol, and PPP authentication and encryption do not meet the security requirements for L2TP. L2TP tunnel authentication provides mutual authentication between the LAC and the LNS at tunnel origination. Therefore, it does not protect control and data traffic on a per packet basis. Thus, L2TP tunnel authentication leaves the L2TP tunnel vulnerable to attacks. PPP authenticates the client to the LNS, but also does not provide per-packet authentication, integrity, or replay protection. PPP encryption meets confidentiality requirements for PPP traffic but does not address authentication, integrity, replay protection and key management requirements. In addition, PPP ECP negotiation, outlined in [10] does not provide for a protected ciphersuite negotiation. Therefore, PPP encryption provides a weak security solution, and in addition does not assist in securing L2TP control channel.

Key management facilities are not provided by the L2TP protocol. However, where L2TP tunnel authentication is desired, it is necessary to distribute tunnel passwords.

Note that several of the attacks outlined above can be carried out on PPP packets sent over the link between the client and the NAS/LAC, prior to encapsulation of the packets within an L2TP tunnel. While strictly speaking these attacks are outside the scope of L2TP security, in order to protect against them, the client SHOULD provide for confidentiality, authentication, replay and integrity protection for PPP packets sent over the dial-up link. Authentication, replay and integrity protection are not currently supported by PPP encryption methods, described in [11]-[13].

2.1. L2TP Security Protocol

The L2TP security protocol MUST provide authentication, integrity and replay protection for control packets. In addition, it SHOULD protect confidentiality of control packets. It MUST provide integrity and replay protection of data packets, and MAY protect confidentiality of data packets. An L2TP security protocol MUST also provide a scalable approach to key management.

To meet the above requirements, all L2TP security compliant implementations MUST implement IPsec ESP for securing both L2TP control and data packets. Transport mode MUST be supported; tunnel mode MAY be supported. All the IPsec-mandated ciphersuites (described in RFC 2406 [4] and RFC 2402 [3]), including NULL encryption MUST be supported. Note that although an implementation MUST support all IPsec ciphersuites, it is an operator choice which ones will be used. If confidentiality is not required (e.g., L2TP data traffic), ESP with NULL encryption may be used. The implementations MUST implement replay protection mechanisms of IPsec.

L2TP security MUST meet the key management requirements of the IPsec protocol suite. IKE SHOULD be supported for authentication, security association negotiation, and key management using the IPsec DOI [5].

2.2. Stateless compression and encryption

Stateless encryption and/or compression is highly desirable when L2TP is run over IP. Since L2TP is a connection-oriented protocol, use of stateful compression/encryption is feasible, but when run over IP, this is not desirable. While providing better compression, when used without an underlying reliable delivery mechanism, stateful methods magnify packet losses. As a result, they are problematic when used over the Internet where packet loss can be significant. Although L2TP [1] is connection oriented, packet ordering is not mandatory,

which can create difficulties in implementation of stateful compression/encryption schemes. These considerations are not as important when L2TP is run over non-IP media such as IEEE 802, ATM, X.25, or Frame Relay, since these media guarantee ordering, and packet losses are typically low.

3. L2TP/IPsec inter-operability guidelines

The following guidelines are established to meet L2TP security requirements using IPsec in practical situations.

3.1. L2TP tunnel and Phase 1 and 2 SA teardown

Mechanisms within PPP and L2TP provide for both graceful and non-graceful teardown. In the case of PPP, an LCP TermReq and TermAck sequence corresponds to a graceful teardown. LCP keep-alive messages and L2TP tunnel hellos provide the capability to detect when a non-graceful teardown has occurred. Whenever teardown events occur, causing the tunnel to close, the control connection teardown mechanism defined in [1] must be used. Once the L2TP tunnel is deleted by either peer, any phase 1 and phase 2 SA's which still exist as a result of the L2TP tunnel between the peers SHOULD be deleted. Phase 1 and phase 2 delete messages SHOULD be sent when this occurs.

When IKE receives a phase 1 or phase 2 delete message, IKE should notify L2TP this event has occurred. If the L2TP state is such that a ZLB ack has been sent in response to a STOPCCN, this can be assumed to be positive acknowledgment that the peer received the ZLB ack and has performed a teardown of any L2TP tunnel state associated with the peer. The L2TP tunnel state and any associated filters can now be safely removed.

3.2. Fragmentation Issues

Since the default MRU for PPP connections is 1500 bytes, fragmentation can become a concern when prepending L2TP and IPsec headers to a PPP frame. One mechanism which can be used to reduce this problem is to provide PPP with the MTU value of the ingress/egress interface of the L2TP/IPsec tunnel minus the overhead of the extra headers. This should occur after the L2TP tunnel has been setup and but before LCP negotiations begin. If the MTU value of the ingress/egress interface for the tunnel is less than PPP's default MTU, it may replace the value being used. This value may also be used as the initial value proposed for the MRU in the LCP config req.

If an ICMP PMTU is received by IPsec, this value should be stored in the SA as proposed in [6]. IPsec should also provide notification of this event to L2TP so that the new MTU value can be reflected into the PPP interface. Any new PTMU discoveries seen at the PPP interface should be checked against this new value and processed accordingly.

3.3. Per-packet security checks

When a packet arrives from a tunnel which requires security, L2TP MUST:

[1] Check to ensure that the packet was decrypted and/or authenticated by IPsec. Since IPsec already verifies that the packet arrived in the correct SA, L2TP can be assured that the packet was indeed sent by a trusted peer and that it did not arrive in the clear.

[2] Verify that the IP addresses and UDP port values in the packet match the socket information which was used to setup the L2TP tunnel. This step prevents malicious peers from spoofing packets into other tunnels.

4. IPsec Filtering details when protecting L2TP

Since IKE/IPsec is agnostic about the nuances of the application it is protecting, typically no integration is necessary between the application and the IPsec protocol. However, protocols which allow the port number to float during the protocol negotiations (such as L2TP), can cause problems within the current IKE framework. The L2TP specification [1] states that implementations MAY use a dynamically assigned UDP source port. This port change is reflected in the SCCRP sent from the responder to the initiator.

Although the current L2TP specification allows the responder to use a new IP address when sending the SCCRP, implementations requiring protection of L2TP via IPsec SHOULD NOT do this. To allow for this behavior when using L2TP and IPsec, when the responder chooses a new IP address it MUST send a StopCCN to the initiator, with the Result and Error Code AVP present. The Result Code MUST be set to 2 (General Error) and the Error Code SHOULD be set to 7 (Try Another). If the Error Code is set to 7, then the optional error message MUST be present and the contents MUST contain the IP address (ASCII encoded) that the Responder desires to use for subsequent communications. Only the ASCII encoded IP address should be present in the error message. The IP address is encoded in dotted decimal format for IPv4 or in RFC 2373 [17] format for IPv6. The initiator MUST parse the result and error code information and send a new SCCRQ

to the new IP address contained in the error message. This approach reduces complexity since now the initiator always knows precisely the IP address of its peer. This also allows a controlled mechanism for L2TP to tie IPsec filters and policy to the same peer.

The filtering details required to accommodate this behavior as well as other mechanisms needed to protect L2TP with IPsec are discussed in the following sections.

4.1. IKE Phase 1 Negotiations

Per IKE [7], when using pre-shared key authentication, a key must be present for each peer to which secure communication is required. When using Main Mode (which provides identity protection), this key must correspond to the IP address for the peer. When using Aggressive Mode (which does not provide identity protection), the pre-shared key must map to one of the valid id types defined in the IPsec DOI [5].

If the initiator receives a StopCCN with the result and error code AVP set to "try another" and a valid IP address is present in the message, it MAY bind the original pre-shared key used by IKE to the new IP address contained in the error-message.

One may may wish to consider the implications for scalability of using pre-shared keys as the authentication method for phase 1. As the number of LAC and LNS endpoints grow, pre-shared keys become increasingly difficult to manage. Whenever possible, authentication with certificates is preferred.

4.2. IKE Phase 2 Negotiations

During the IKE phase 2 negotiations, the peers agree on what traffic is to be protected by the IPsec protocols. The quick mode IDs represent the traffic which the peers agree to protect and are comprised of address space, protocol, and port information.

When securing L2TP with IPsec, the following cases must be considered:

Cases:

Initiator Port	Responder Addr	Responder Port
1701	Fixed	1701
1701	Fixed	Dynamic
1701	Dynamic	1701
1701	Dynamic	Dynamic
Dynamic	Fixed	1701
Dynamic	Fixed	Dynamic
Dynamic	Dynamic	1701
Dynamic	Dynamic	Dynamic

By solving the most general case of the above permutations, all cases are covered. The most general case is the last one in the list. This scenario is when the initiator chooses a new port number and the responder chooses a new address and port number. The L2TP message flow which occurs to setup this sequence is as follows:

-> IKE Phase 1 and Phase 2 to protect Initial SCCRQ

```
        SCCRQ ->         (Fixed IP address, Dynamic Initiator Port)
              <- STOPCCN (Responder chooses new IP address)
```

-> New IKE Phase 1 and Phase 2 to protect new SCCRQ

```
        SCCRQ ->         (SCCRQ to Responder's new IP address)
```

<- New IKE Phase 2 to for port number change by the responder

```
              <- SCCRP   (Responder chooses new port number)
        SCCCN ->         (L2TP Tunnel Establishment completes)
```

Although the Initiator and Responder typically do not dynamically change ports, L2TP security must accommodate emerging applications such as load balancing and QoS. This may require that the port and IP address float during L2TP tunnel establishment.

To support the general case, mechanisms must be designed into L2TP and IPsec which allow L2TP to inject filters into the IPsec filter database. This technique may be used by any application which floats ports and requires security via IPsec, and is described in the following sections.

The responder is not required to support the ability to float its IP address and port. However, the initiator MUST allow the responder to float its port and SHOULD allow the responder to choose a new IP address (see section 4.2.3, below).

Appendix A provides examples of these cases using the process described below.

4.2.1. Terminology definitions used for filtering statements

I-Port The UDP port number the Initiator chooses to originate/receive L2TP traffic on. This can be a static port such as 1701 or an ephemeral one assigned by the socket.

R-Port The UDP port number the Responder chooses to originate/receive L2TP traffic on. This can be the port number 1701 or an ephemeral one assigned by the socket. This is the port number the Responder uses after receiving the initial SCCRQ.

R-IPAddr1 The IP address the Responder listens on for initial SCCRQ. If the responder does not choose a new IP address, this address will be used for all subsequent L2TP traffic.

R-IPAddr2 The IP address the Responder chooses upon receiving the SCCRQ. This address is used to send the SCCRP and all subsequent L2TP tunnel traffic is sent and received on this address.

R-IPAddr The IP address which the responder uses for sending and receiving L2TP packets. This is either the initial value of R-IPAddr1 or a new value of R-IPAddr2.

I-IPAddr The IP address the Initiator uses to communicate with for the L2TP tunnel.

Any-Addr The presence of Any-Address defines that IKE should accept any single address proposed in the local address of the quick mode IDs sent by the peer during IKE phase 2 negotiations. This single address may be formatted as an

IP Single address, an IP Netmask address with the Netmask set to 255.255.255.255, and IP address Range with the range being 1, or a hostname which can be resolved to one address. Refer to [5] for more information on the format for quick mode IDs.

Any-Port The presence of Any-Port defines that IKE should accept a value of 0 or a specific port value for the port value in the port value in the quick mode IDs negotiated during IKE phase 2.

The filters defined in the following sections are listed from highest priority to lowest priority.

4.2.2. Initial filters needed to protect the SCCRQ

The initial filter set on the initiator and responder is necessary to protect the SCCRQ sent by the initiator to open the L2TP tunnel. Both the initiator and the responder must either be pre-configured for these filters or L2TP must have a method to inject this information into the IPsec filtering database. In either case, this filter MUST be present before the L2TP tunnel setup messages start to flow.

Responder Filters:
Outbound-1: None. They should be be dynamically created by IKE upon successful completion of phase 2.

Inbound-1: From Any-Addr, to R-IPAddr1, UDP, src Any-Port, dst 1701

Initiator Filters:
Outbound-1: From I-IPAddr, to R-IPAddr1, UDP, src I-Port, dst 1701

Inbound-1: From R-IPAddr1, to I-IPAddr, UDP, src 1701, dst I-Port
Inbound-2: From R-IPAddr1, to I-IPAddr, UDP, src Any-Port, dst I-Port

When the initiator uses dynamic ports, L2TP must inject the filters into the IPsec filter database, once its source port number is known. If the initiator uses a fixed port of 1701, these filters MAY be statically defined.

The Any-Port definition in the initiator’s inbound-2 filter statement is needed to handle the potential port change which may occur as the result of the responder changing its port number.

If a phase 2 SA bundle is not already present to protect the SCCRQ, the sending of a SCCRQ by the initiator SHOULD cause IKE to setup the necessary SAs to protect this packet. Alternatively, L2TP may also request IKE to setup the SA bundle. If the SA cannot be setup for some reason, the packet MUST be dropped.

The port numbers in the Quick Mode IDs sent by the initiator MUST contain the specific port numbers used to identify the UDP socket. The port numbers would be either I-Port/1701 or 1701/1701 for the initial SCCRQ. The quick mode IDs sent by the initiator will be a subset of the Inbound-1 filter at the responder. As a result, the quick mode exchange will finish and IKE should inject a specific filter set into the IPsec filter database and associate this filter set with the phase 2 SA established between the peers. These filters should persist as long as the L2TP tunnel exists. The new filter set at the responder will be:

```
Responder Filters:
   Outbound-1: From R-IPAddr1, to I-IPAddr,  UDP, src 1701,
   dst I-Port

   Inbound-1:  From I-IPAddr,  to R-IPAddr1, UDP, src I-Port,
   dst 1701
   Inbound-2:  From Any-Addr,  to R-IPAddr1, UDP, src Any-Port,
   dst 1701
```

Mechanisms SHOULD exist between L2TP and IPsec such that L2TP is not retransmitting the SCCRQ while the SA is being established. L2TP's control channel retransmit mechanisms should start once the SA has been established. This will help avoid timeouts which may occur as the result of slow SA establishment.

Once the phase 2 SA has been established between the peers, the SCCRQ should be sent from the initiator to the responder.

If the responder does not choose a new IP address or a new port number, the L2TP tunnel can now proceed to establish.

4.2.3. Responder chooses new IP Address

This step describes the process which should be followed when the responder chooses a new IP address. The only opportunity for the responder to change its IP address is after receiving the SCCRQ but before sending a SCCRP.

The new address the responder chooses to use MUST be reflected in the result and error code AVP of a STOPCCN message. The Result Code MUST be set to 2 (General Error) and the Error Code MUST be set to 7 (Try

Another). The optional error message MUST be present and the contents MUST contain the IP address (ASCII encoded) the Responder desires to use for subsequent communications. Only the ASCII encoded IP address should be present in the error message. The IP address is encoded in dotted decimal format for IPv4 or in RFC 2373 [17] format for IPv6.

The STOPCCN Message MUST be sent using the same address and UDP port information which the initiator used to send the SCCRQ. This message will be protecting using the initial SA bundle setup to protect the SCCRQ.

Upon receiving the STOPCCN, the initiator MUST parse the IP address from the Result and Error Code AVP and perform the necessary sanity checks to verify this is a correctly formatted address. If no errors are found L2TP should inject a new set of filters into the IPsec filter database. If using pre-shared key authentication, L2TP MAY request IKE to bind the new IP address to the pre-shared key which was used for the original IP address.

Since the IP address of the responder changed, a new phase 1 and phase 2 SA must be established between the peers before the new SCCRQ is sent.

Assuming the initial tunnel has been torn down and the filters needed to create the tunnel removed, the new filters for the initiator and responder will be:

```
Initiator Filters:
   Outbound-1: From I-IPAddr,  to R-IPAddr2, UDP, src I-Port,
   dst 1701

   Inbound-1:  From R-IPAddr2, to I-IPAddr,  UDP, src 1701,
   dst I-Port
   Inbound-2:  From R-IPAddr2, to I-IPAddr,  UDP, src Any-Port,
   dst I-Port
```

Once IKE phase 2 completes, the new filter set at the responder will be:

```
Responder Filters:
   Outbound-1: From R-IPAddr2, to I-IPAddr,  UDP, src 1701,
   dst I-Port

   Inbound-1:  From I-IPAddr,  to R-IPAddr2, UDP, src I-Port,
   dst 1701
   Inbound-2:  From Any-Addr,  to R-IPAddr1, UDP, src Any-Port,
   dst 1701
```

If the responder chooses not to move to a new port number, the L2TP tunnel setup can now complete.

4.2.4. Responder chooses new Port Number

The responder MAY choose a new UDP source port to use for L2TP tunnel traffic. This decision MUST be made before sending the SCCRP. If a new port number is chosen, then L2TP must inject new filters into the IPsec filter database. The responder must start new IKE phase 2 negotiations with the initiator.

The final filter set at the initiator and responder is as follows.

Initiator Filters:

Outbound-1: From I-IPAddr, to R-IPAddr, UDP, src I-Port, dst R-Port
Outbound-2: From I-IPAddr, to R-IPAddr, UDP, src I-Port, dst 1701

Inbound-1: From R-IPAddr, to I-IPAddr, UDP, src R-Port, dst I-Port
Inbound-2: From R-IPAddr, to I-IPAddr, UDP, src 1701, dst I-Port
Inbound-3: From R-IPAddr, to I-IPAddr, UDP, src Any-Port, dst I-Port

The Inbound-1 filter for the initiator will be injected by IKE upon successful completion of the phase 2 negotiations initiated by the peer.

Responder Filters:

Outbound-1: From R-IPAddr, to I-IPAddr, UDP, src R-Port, dst I-Port
Outbound-2: From R-IPAddr, to I-IPAddr, UDP, src 1701, dst I-Port

Inbound-1: From I-IPAddr, to R-IPAddr, UDP, src I-Port, dst R-Port
Inbound-2: From I-IPAddr, to R-IPAddr, UDP, src I-Port, dst 1701
Inbound-3: From Any-Addr, to R-IPAddr1, UDP, src Any-Port, dst 1701

Once the negotiations have completed, the SCCRP is sent and the L2TP tunnel can complete establishment. After the L2TP tunnel has been established, any residual SAs and their associated filters may be deleted.

4.2.5. Gateway-gateway and L2TP Dial-out considerations

In the gateway-gateway or the L2TP dial-out scenario, either side may initiate L2TP. The process outlined in the previous steps should be followed with one addition. The initial filter set at both sides MUST include the following filter:

```
Inbound Filter:
   1: From Any-Addr, to R-IPAddr1, UDP, src Any-Port, dst 1701
```

When either peer decides to start a tunnel, L2TP should inject the necessary inbound and outbound filters to protect the SCCRQ. Tunnel establishment then proceeds exactly as stated in the previous sections.

5. Security Considerations

5.1. Authentication issues

IPsec IKE negotiation MUST negotiate an authentication method specified in the IKE RFC 2409 [7]. In addition to IKE authentication, L2TP implementations utilize PPP authentication methods, such as those described in [15]-[16]. In this section, we discuss authentication issues.

5.1.1. Differences between IKE and PPP authentication

While PPP provides initial authentication, it does not provide per-packet authentication, integrity or replay protection. This implies that the identity verified in the initial PPP authentication is not subsequently verified on reception of each packet.

With IPsec, when the identity asserted in IKE is authenticated, the resulting derived keys are used to provide per-packet authentication, integrity and replay protection. As a result, the identity verified in the IKE conversation is subsequently verified on reception of each packet.

Let us assume that the identity claimed in PPP is a user identity, while the identity claimed within IKE is a machine identity. Since only the machine identity is verified on a per-packet basis, there is no way to verify that only the user authenticated within PPP is using the tunnel. In fact, IPsec implementations that only support machine authentication typically have no way to enforce traffic segregation. As a result, where machine authentication is used, once an L2TP/IPsec tunnel is opened, any user on a multi-user machine will typically be able to send traffic down the tunnel.

If the IPsec implementation supports user authentication, this problem can be averted. In this case, the user identity asserted within IKE will be verified on a per-packet basis. In order to provide segregation of traffic between users when user authentication is used, the client MUST ensure that only traffic from that particular user is sent down the L2TP tunnel.

5.1.2. Certificate authentication in IKE

When X.509 certificate authentication is chosen within IKE, the LNS is expected to use an IKE Certificate Request Payload (CRP) to request from the client a certificate issued by a particular certificate authority or may use several CRPs if several certificate authorities are trusted and configured in its IPsec IKE authentication policy.

The LNS SHOULD be able to trust several certificate authorities in order to allow tunnel client end-points to connect to it using their own certificate credential from their chosen PKI. Client and server side certificate revocation list checking MAY be enabled on a per-CA basis, since differences in revocation list checking exist between different PKI providers.

L2TP implementations MAY use dynamically assigned ports for both source and destination ports only if security for each source and destination port combination can be successfully negotiated by IKE.

5.1.3. Machine versus user certificate authentication in IKE

The certificate credentials provided by the L2TP client during the IKE negotiation MAY be those of the machine or of the L2TP user. When machine authentication is used, the machine certificate is typically stored on the LAC and LNS during an enrollment process. When user certificates are used, the user certificate can be stored either on the machine or on a smartcard.

Since the value of a machine certificate is inversely proportional to the ease with which an attacker can obtain one under false pretenses, it is advisable that the machine certificate enrollment process be strictly controlled. For example, only administrators may have the ability to enroll a machine with a machine certificate.

While smartcard certificate storage lessens the probability of compromise of the private key, smartcards are not necessarily desirable in all situations. For example, some organizations deploying machine certificates use them so as to restrict use of non-approved hardware. Since user authentication can be provided

within PPP (keeping in mind the weaknesses described earlier), support for machine authentication in IPsec makes it is possible to authenticate both the machine as well as the user.

In circumstances in which this dual assurance is considered valuable, enabling movement of the machine certificate from one machine to another, as would be possible if the machine certificate were stored on a smart card, may be undesirable.

Similarly, when user certificate are deployed, it is advisable for the user enrollment process to be strictly controlled. If for example, a user password can be readily used to obtain a certificate (either a temporary or a longer term one), then that certificate has no more security value than the password. To limit the ability of an attacker to obtain a user certificate from a stolen password, the enrollment period can be limited, after which password access will be turned off. Such a policy will prevent an attacker obtaining the password of an unused account from obtaining a user certificate once the enrollment period has expired.

5.1.4. Pre-shared keys in IKE

Use of pre-shared keys in IKE main mode is vulnerable to man-in-the-middle attacks when used in remote access situations. In main mode it is necessary for SKEYID_e to be used prior to the receipt of the identification payload. Therefore the selection of the pre-shared key may only be based on information contained in the IP header. However, in remote access situations, dynamic IP address assignment is typical, so that it is often not possible to identify the required pre-shared key based on the IP address.

Thus when pre-shared keys are used in remote access scenarios, the same pre-shared key is shared by a group of users and is no longer able to function as an effective shared secret. In this situation, neither the client nor the server identifies itself during IKE phase 1; it is only known that both parties are a member of the group with knowledge of the pre-shared key. This permits anyone with access to the group pre-shared key to act as a man-in-the-middle.

This vulnerability does not occur in aggressive mode since the identity payload is sent earlier in the exchange, and therefore the pre-shared key can be selected based on the identity. However, when aggressive mode is used the user identity is exposed and this is often considered undesirable.

As a result, where main mode is used with pre-shared keys, unless PPP performs mutual authentication, the server is not authenticated. This enables a rogue server in possession of the group pre-shared key

to successfully masquerade as the LNS and mount a dictionary attack on legacy authentication methods such as CHAP [15]. Such an attack could potentially compromise many passwords at a time. This vulnerability is present in some existing IPsec tunnel mode implementations.

To avoid this problem, L2TP/IPsec implementations SHOULD NOT use a group pre-shared key for IKE authentication to the LNS. IKE pre-shared authentication key values SHOULD be protected in a manner similar to the user's account password used by L2TP.

5.2. IPsec and PPP security interactions

When L2TP is protected with IPsec, both PPP and IPsec security services are available. Which services are negotiated depends on whether the tunnel is compulsory or voluntary. A detailed analysis of voluntary and compulsory tunneling scenarios is included below. These scenarios are non-normative and do not create requirements for an implementation to be L2TP security compliant.

In the scenarios below, it is assumed that both L2TP clients and servers are able to set and get the properties of IPsec security associations, as well as to influence the IPsec security services negotiated. Furthermore, it is assumed that L2TP clients and servers are able to influence the negotiation process for PPP encryption and compression.

5.2.1. Compulsory tunnel

In the case of a compulsory tunnel, the client sends PPP frames to the LAC, and will typically not be aware that the frames are being tunneled, nor that any security services are in place between the LAC and LNS. At the LNS, a data packet will arrive, which includes a PPP frame encapsulated in L2TP, which is itself encapsulated in an IP packet. By obtaining the properties of the Security Association set up between the LNS and the LAC, the LNS can obtain information about security services in place between itself and the LAC. Thus in the compulsory tunneling case, the client and the LNS have unequal knowledge of the security services in place between them.

Since the LNS is capable of knowing whether confidentiality, authentication, integrity and replay protection are in place between itself and the LAC, it can use this knowledge in order to modify its behavior during PPP ECP [10] and CCP [9] negotiation. Let us assume that LNS confidentiality policy can be described by one of the following terms: "Require Encryption," "Allow Encryption" or "Prohibit Encryption." If IPsec confidentiality services are in place, then an LNS implementing a "Prohibit Encryption" policy will

act as though the policy had been violated. Similarly, an LNS implementing a "Require Encryption" or "Allow Encryption" policy will act as though these policies were satisfied, and would not mandate use of PPP encryption or compression. This is not the same as insisting that PPP encryption and compression be turned off, since this decision will depend on client policy.

Since the client has no knowledge of the security services in place between the LAC and the LNS, and since it may not trust the LAC or the wire between itself and the LAC, the client will typically want to ensure sufficient security through use of end-to-end IPsec or PPP encryption/compression between itself and the LNS.

A client wishing to ensure security services over the entire travel path would not modify this behavior even if it had knowledge of the security services in place between the LAC and the LNS. The client negotiates confidentiality services between itself and the LNS in order to provide privacy on the wire between itself and the LAC. The client negotiates end-to-end security between itself and the end-station in order to ensure confidentiality on the portion of the path between the LNS and the end-station.

The client will typically not trust the LAC and will negotiate confidentiality and compression services on its own. As a result, the LAC may only wish to negotiate IPsec ESP with null encryption with the LNS, and the LNS will request replay protection. This will ensure that confidentiality and compression services will not be duplicated over the path between the LAC and the LNS. This results in better scalability for the LAC, since encryption will be handled by the client and the LNS.

The client can satisfy its desire for confidentiality services in one of two ways. If it knows that all end-stations that it will communicate with are IPsec-capable (or if it refuses to talk to non-IPsec capable end-stations), then it can refuse to negotiate PPP encryption/compression and negotiate IPsec ESP with the end-stations instead. If the client does not know that all end-stations it will contact are IPsec capable (the most likely case), then it will negotiate PPP encryption/compression. This may result in duplicate compression/encryption which can only be eliminated if PPP compression/encryption can be turned off on a per-packet basis. Note that since the LNS knows that the client’s packets are being tunneled but the client does not, the LNS can ensure that stateless compression/encryption is used by offering stateless compression/encryption methods if available in the ECP and CCP negotiations.

5.2.2. Voluntary tunnel

In the case of a voluntary tunnel, the client will be send L2TP packets to the NAS, which will route them to the LNS. Over a dialup link, these L2TP packets will be encapsulated in IP and PPP. Assuming that it is possible for the client to retrieve the properties of the Security Association between itself and the LNS, the client will have knowledge of any security services negotiated between itself and the LNS. It will also have knowledge of PPP encryption/compression services negotiated between itself and the NAS.

From the LNS point of view, it will note a PPP frame encapsulated in L2TP, which is itself encapsulated in an IP packet. This situation is identical to the compulsory tunneling case. If LNS retrieves the properties of the Security Association set up between itself and the client, it can be informed of the security services in place between them. Thus in the voluntary tunneling case, the client and the LNS have symmetric knowledge of the security services in place between them.

Since the LNS is capable of knowing whether confidentiality, authentication, integrity check or replay protection is in place between the client and itself, it is able to use this knowledge to modify its PPP ECP and CCP negotiation stance. If IPsec confidentiality is in place, the LNS can behave as though a "Require Encryption" directive had been fulfilled, not mandating use of PPP encryption or compression. Typically the LNS will not insist that PPP encryption/compression be turned off, instead leaving this decision to the client.

Since the client has knowledge of the security services in place between itself and the LNS, it can act as though a "Require Encryption" directive had been fulfilled if IPsec ESP was already in place between itself and the LNS. Thus, it can request that PPP encryption and compression not be negotiated. If IP compression services cannot be negotiated, it will typically be desirable to turn off PPP compression if no stateless method is available, due to the undesirable effects of stateful PPP compression.

Thus in the voluntary tunneling case the client and LNS will typically be able to avoid use of PPP encryption and compression, negotiating IPsec Confidentiality, Authentication, and Integrity protection services instead, as well as IP Compression, if available.

This may result in duplicate encryption if the client is communicating with an IPsec-capable end-station. In order to avoid duplicate encryption/compression, the client may negotiate two

Security Associations with the LNS, one with ESP with null encryption, and one with confidentiality/compression. Packets going to an IPsec- capable end-station would run over the ESP with null encryption security association, and packets to a non-IPsec capable end-station would run over the other security association. Note that many IPsec implementations cannot support this without allowing L2TP packets on the same tunnel to be originated from multiple UDP ports. This requires modifications to the L2TP specification.

Also note that the client may wish to put confidentiality services in place for non-tunneled packets traveling between itself and the NAS. This will protect the client against eavesdropping on the wire between itself and the NAS. As a result, it may wish to negotiate PPP encryption and compression with the NAS. As in compulsory tunneling, this will result in duplicate encryption and possibly compression unless PPP compression/encryption can be turned off on a per-packet basis.

6. References

[1] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol L2TP", RFC 2661, August 1999.

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

[3] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[4] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[5] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", RFC 2407, November 1998.

[6] Atkinson, R. and S. Kent, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

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

[8] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[9] Rand, D., "The PPP Compression Control Protocol (CCP)", RFC 1962, June 1996.

[10] Meyer, G., "The PPP Encryption Control Protocol (ECP)", RFC 1968, June 1996.

[11] Sklower, K. and G. Meyer, "The PPP DES Encryption Protocol (DESE)", RFC 1969, June 1996.

[12] Sklower, K. and G. Meyer, "The PPP DES Encryption Protocol, Version 2 (DESE-bis)", RFC 2419, September 1998.

[13] Hummert, K., "The PPP Triple-DES Encryption Protocol (3DESE)", RFC 2420, September 1998.

[14] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, November 1998.

[15] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, August 1996.

[16] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," RFC 2284, March 1998.

[17] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

Acknowledgments

Thanks to Gurdeep Singh Pall, David Eitelbach, Peter Ford, and Sanjay Anand of Microsoft, John Richardson of Intel and Rob Adams of Cisco for useful discussions of this problem space.

Authors' Addresses

Baiju V. Patel
Intel Corp
2511 NE 25th Ave
Hillsboro, OR 97124

Phone: +1 503 702 2303
EMail: baiju.v.patel@intel.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 706-6605
EMail: bernarda@microsoft.com

William Dixon
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 703 8729
EMail: wdixon@microsoft.com

Glen Zorn
Cisco Systems, Inc.
500 108th Avenue N.E., Suite 500
Bellevue, Washington 98004

Phone: +1 425 438 8218
Fax: +1 425 438 1848
EMail: gwz@cisco.com

Skip Booth
Cisco Systems
7025 Kit Creek Road
RTP, NC 27709

Phone: +1 919 392 6951
EMail: ebooth@cisco.com

Appendix A: Example IPsec Filter sets for L2TP Tunnel Establishment

This section provides examples of IPsec filter sets for L2TP tunnel establishment. While example filter sets are for IPv4, similar examples could just as easily be constructed for IPv6.

A.1 Initiator and Responder use fixed addresses and ports

This is the most simple of the cases since nothing changes during L2TP tunnel establishment. Since the initiator does not know whether the responder will change its port number, it still must be prepared for this case. In this example, the initiator will use an IPv4 address of 1.1.1.1 and the responder will use an IPv4 address of 2.2.2.1.

The filters for this scenario are:

A.1.1 Protect the SCCRQ

```
Initiator Filters:
   Outbound-1: From 1.1.1.1, to 2.2.2.1, UDP, src 1701,     dst 1701

   Inbound-1:  From 2.2.2.1, to 1.1.1.1, UDP, src 1701,     dst 1701
   Inbound-2:  From 2.2.2.1, to 1.1.1.1, UDP, src Any-Port, dst 1701

Responder Filters:
   Outbound-1: None, dynamically injected when IKE Phase 2 completes

   Inbound-1:  From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701
```

After IKE Phase 2 completes the filters at the initiator and responder will be:

```
Initiator Filters:
   Outbound-1: From 1.1.1.1, to 2.2.2.1, UDP, src 1701,     dst 1701

   Inbound-1:  From 2.2.2.1, to 1.1.1.1, UDP, src 1701,     dst 1701
   Inbound-2:  From 2.2.2.1, to 1.1.1.1, UDP, src Any-Port, dst 1701

Responder Filters:
   Outbound-1: From 2.2.2.1,  to 1.1.1.1, UDP, src 1701,     dst 1701

   Inbound-1:  From 1.1.1.1,  to 2.2.2.1, UDP, src 1701,     dst 1701
   Inbound-2:  From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701
```

A.2 Gateway to Gateway Scenario where Initiator and Responder use dynamic ports

In this scenario either side is allowed to initiate the tunnel. Since dynamic ports will be used, an extra phase 2 negotiation must occur to protect the SCCRP sent from the responder to the initiator. Other than the additional phase 2 setup, the only other difference is that L2TP on the responder must inject an additional filter into the IPsec database once the new port number is chosen.

This example also shows the additional filter needed by the initiator which allows either side to start the tunnel. In either the dial-out or the gateway to gateway scenario this additional filter is required.

For this example, assume the dynamic port given to the initiator is 5000 and his IP address is 1.1.1.1. The responder will use an IP address of 2.2.2.1 and a port number of 6000.

The filters for this scenario are:

A.2.1 Initial Filters to allow either side to respond to negotiations

In this case both peers must be able to accept phase 2 negotiations to from L2TP peers. My-IPAddr is defined as whatever IP address the device is willing to accept L2TP negotiations on.

Responder Filters present at both peers:
Inbound-1: From Any-Addr, to My-IPAddr, UDP, src Any-Port, dst 1701

Note: The source IP in the inbound-1 filter above for gateway to gateway tunnels can be IP specific, such as 1.1.1.1, not necessarily Any-Addr.

A.2.2 Protect the SCCRQ, one peer is now the initiator

Initiator Filters:
Outbound-1: From 1.1.1.1, to 2.2.2.1, UDP, src 5000, dst 1701

Inbound-1: From 2.2.2.1, to 1.1.1.1, UDP, src 1701, dst 5000
Inbound-2: From 2.2.2.1, to 1.1.1.1, UDP, src Any-Port, dst 5000
Inbound-3: From Any-Addr, to 1.1.1.1, UDP, src Any-Port, dst 1701

Responder Filters:
Outbound-1: None, dynamically injected when IKE Phase 2 completes

Inbound-1: From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701

After IKE Phase 2 completes the filters at the initiator and responder will be:

```
Initiator Filters:
   Outbound-1: From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 1701

   Inbound-1:  From 2.2.2.1,  to 1.1.1.1, UDP, src 1701,     dst 5000
   Inbound-2:  From 2.2.2.1,  to 1.1.1.1, UDP, src Any-Port, dst 5000

   Inbound-3:  From Any-Addr, to 1.1.1.1, UDP, src Any-Port, dst 1701

Responder Filters:
   Outbound-1: From 2.2.2.1,  to 1.1.1.1, UDP, src 1701,     dst 5000

   Inbound-1:  From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 1701
   Inbound-2:  From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701
```

A.2.3 Protect the SCCRP after port change

At this point the responder knows which port number it is going to use. New filters should be injected by L2TP to reflect this new port assignment.

The new filter set at the responder is:

```
Responder Filters:
   Outbound-1: From 2.2.2.1,  to 1.1.1.1, UDP, src 6000,     dst 5000
   Outbound-2: From 2.2.2.1,  to 1.1.1.1, UDP, src 1701,     dst 5000

   Inbound-1:  From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 6000
   Inbound-2:  From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 1701
   Inbound-3:  From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701
```

The second phase 2 will start once L2TP sends the SCCRP. Once the phase 2 negotiations complete, the new filter set at the initiator and the responder will be:

```
Initiator Filters:
   Outbound-1: From 1.1.1.1, to 2.2.2.1, UDP, src 5000,     dst 6000
   Outbound-2: From 1.1.1.1, to 2.2.2.1, UDP, src 5000,     dst 1701

   Inbound-1:  From 2.2.2.1, to 1.1.1.1, UDP, src 6000,     dst 5000
   Inbound-2:  From 2.2.2.1, to 1.1.1.1, UDP, src 1701,     dst 5000
   Inbound-3:  From 2.2.2.1, to 1.1.1.1, UDP, src Any-Port, dst 1701
```

```
Responder Filters:
   Outbound-1: From 2.2.2.1,  to 1.1.1.1, UDP, src 6000,     dst 5000
   Outbound-2: From 2.2.2.1,  to 1.1.1.1, UDP, src 1701,     dst 5000

   Inbound-1:  From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 6000
   Inbound-2:  From 1.1.1.1,  to 2.2.2.1, UDP, src 5000,     dst 1701
   Inbound-3:  From Any-Addr, to 2.2.2.1, UDP, src Any-Port, dst 1701
```

Once the L2TP tunnel has been successfully established, the original phase 2 may be deleted. This allows the Inbound-2 and Outbound-2 filter statements to be removed as well.

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.