Manual SA Configuration for IPv6 Link Local Messages

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. 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 may be found at http://www.ietf.org/ietf/1id-abstracts.txt

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

The distribution of this memo is unlimited. It is filed as <draft-arkko-manual-icmpv6-sas-00.txt>, and expires August 1, 2001. Please send comments to the author or to IPsec and/or IPNG working group mailing lists.

2. Abstract

This draft discusses the use of manually configured IPsec SAs to protect ICMPv6 messages such as router discovery and address resolution on the local link. IPsec SAs are generally identified by the triple <SPI, destination address, protocol>. For the ICMPv6 messages configuring the SAs requires some effort, however, since there are multiple known destination addresses plus a number of addresses that depend on the physical link addresses. This draft describes the security implications of protecting or not protecting the link local ICMPv6 messages, lists the SAs that must be configured manually, and discusses some approaches for reducing configuration effort.

3. Introduction

IPv6 architecture makes it possible to secure all IP packets using IPsec [6], even control messages and even to multicast addresses. IPsec architecture has a Security Policy Database that specifies which
traffic is protected, and how.

The control message protocol for IPv6, ICMPv6 [3], contains many messages that are intended to be used for router discovery, address resolution, and other tasks on the local LAN link. As described elsewhere [4], dynamically negotiated SAs cannot be used to protect these messages as the negotiation would already require the messages to have been exchanged. Instead, manually configured SAs must be used. However, setting up these SA is not easy. This draft lists those fixed multicast, node addresses, and node address dependent multicast addresses for which the SAs must be created. We will also discuss the security implications of the link local ICMPv6 messages under various assumptions of attacker capabilities. Finally, we discuss approaches that can be used to reduce configuration work in setting up the manual SAs. These approaches include configuration tools, the use of well-known SPI numbers with special treatment, and new protocols.

4. Nature of the Addresses

ICMPv6 messages are sent using various kinds of source and destination address types. As the destination address is used to find the right SA, the nature of the destination address is of relevance here. The destination address can be either a well known multicast address, a computed multicast address, such as the solicited-node multicast address, or a unicast address. While many ICMPv6 messages use multicast addresses most of the time, some also use unicast addresses sometimes. For instance, the Neighbour Solicitation messages are usually sent to multicast addresses, but the Neighbour Advertisement messages are also sent to unicast addresses when sent as a response to a node that has an address.

Unfortunately, the ICMPv6 specifications do not always use very specific language when talking about the possible forms of addresses for the messages. The word "typically" is often used to describe an address that would most logically be used for a particular message, but a question remains whether other addresses could be used in some situations or in some implementations.

5. ICMPv6 Tasks

In IPv6, ICMP has several tasks, and many of these tasks are overloaded on a few central message types such as the Neighbour Discovery message. In this chapter we explain some of these tasks and their effects in order to understand better how the messages should be treated. We will only discuss those tasks for which manual SA protection is relevant, i.e., which cannot be protected using IKE-based [7] SAs according to [4].

5.1. Router and Prefix Discovery

Router and prefix discovery is a part of the Neighbour Discovery protocol [1], which in turn is a part of the ICMPv6. The main purpose of the router discovery is to find neighboring routers that are willing to forward packets on the behalf of hosts. Prefix discovery involves
determining which destinations are directly on a link; this information is necessary in order to know whether a packet should be sent to a router or to the destination directly. Typically, address autoconfiguration and other tasks can’t proceed until the router discovery process has run.

The Router Solicitation and Router Advertisement messages are used for this and only this purpose.

The Router Solicitation message has ICMPv6 type 133. The destination address is typically the All Routers multicast address [1]. This message is always used only locally on the link.

The Router Advertisement message has ICMPv6 type 134. The destination address can be either a unicast or the All Nodes multicast address [1]. Like the solicitation message, the advertisement is also local to the link only.

5.2. Address Autoconfiguration

Address autoconfiguration is another part of the Neighbour Discovery protocol [1, 2]. Its purpose is to automatically assign addresses to interfaces. It comes in two variants, stateless and statefull. In this document we consider only the stateless autoconfiguration aspects.

The Neighbour Solicitation and Advertisement messages are used for this purpose, among other things. Furthermore, Router and Prefix Discovery and Duplicate Address Detection have an effect to the Address Autoconfiguration tasks.

The Neighbour Solicitation message has ICMPv6 type 135. The destination address is either the solicited node multicast address, unicast address, or an anycast address [1, 3]. Neighbour Solicitation and Advertisement messages are used for multiple purposes: address autoconfiguration, duplicate address detection, and reachability detection. In all these roles they are link local.

The Neighbour Advertisement type is 136. The the destination address is either a unicast address or the All Nodes multicast address [1]. Like the solicitation message, this message is link local.

5.3. Duplicate Address Detection

As a part of the stateless address autoconfiguration procedure, nodes check for duplicate addresses prior to assigning an address to an interface [2]. This procedure uses the same messages as the Neighbour Discovery protocol [1]. Since the rules outlined in [2] forbid the use of an address for both sending and receiving packets until it has been found unique, no higher layer traffic is possible until this procedure has completed.

The Neighbour Solicitation and Advertisement messages are used also for this purpose.
5.4. Address Resolution

In address resolution nodes determine the link-layer address of a local destination given only the destination’s IP address [1]. Again, no higher level traffic can proceed until the sender knows the hardware address of the destination or the next hop router.

The Neighbour Solicitation and Advertisement messages are used also for this purpose.

5.5. Neighbor Reachability Detection

Hosts monitor the reachability of local destinations and routers in the Neighbour Unreachability procedure, which is a part of the Neighbour Discovery protocol [1]. No higher level traffic can proceed if this procedure flushes out neighbour cache entries after (perhaps incorrectly) determining that the peer is not reachable.

The Neighbour Solicitation and Advertisement messages are used also for this purpose.

5.6. Redirect

In the Redirect procedure, a router informs a host of a better first-hop node to reach a particular destination [1]. It is a part of the Neighbour Discovery protocol. As routers forward packets regardless of them being sent first to the wrong place, communications can still be established without the ability to process Redirect messages.

The Redirect message is used solely in the Redirect procedure. Its ICMPv6 type is 137. The Redirect message is always sent from a unicast addresses to the source address of the packet that triggered the Redirect [1]. Rules in [5] dictate that unspecified, anycast, or multicast addresses may not be used as source addresses. Therefore, the destination address will always be a unicast address. This message is only used for link local purposes, not for end-to-end communications.

6. Recommendation for manual SA setup

As described above, ICMPv6 uses as destination address unicast addresses, known multicast addresses, and a set of computed multicast addresses (based on the physical link addresses). On a given link, the following SAs are required if all ICMPv6 traffic is to be secured with IPsec:

(1) All Nodes multicast address FF02:0:0:0:0:0:0:1 [5].

(2) All Routers multicast address FF02:0:0:0:0:0:0:1 [5].

(3) For each node on the network, the Solicited Node multicast address FF02:0:0:0:0:1:FFXX:XXXX [5]. Note that, in general, there will be multiple such addresses when there are multiple nodes, but it is also possible that some of these multicast addresses collide since only 24 bits from the physical link addresses are used in the multicast
address.

(4) For each node on the network, the unicast address of that node.

(5) For each node that has multiple unicast addresses, items (3) and (4) have to be repeated for each address.

These rules result in at most $2 \times (N \times M + 1)$ SAs where $N$ is the number of nodes in the network and $M$ is the number of addresses each node has. Table 1 below tabulates the number of SAs required as a function of the number of nodes in the network:

<table>
<thead>
<tr>
<th># of Nodes</th>
<th># of Addrs</th>
<th># of SAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>82</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>102</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>202</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>202</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>402</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>2002</td>
</tr>
<tr>
<td>1000</td>
<td>2</td>
<td>4002</td>
</tr>
</tbody>
</table>

The above setup also implies that one knows beforehand the physical link and IP level addresses of each node on the link, including the multiple addresses mentioned under (5). Normally, a node may not necessarily know the addresses itself but they are rather specified in the set of router advertisements received. These things complicate the setup but is similar to the requirement that the keys can be configured manually for all of these nodes; it isn’t possible to provide security and at the same time allow previously unknown nodes to the system when manual keying is used.
Note that if the privacy extension [9] is used for address autoconfiguration, it may not be possible to know the physical address identifiers. Even if the hosts could reveal this information for some time to the future, the network would have to support multiple addresses for the same node. One possible way to deal with this is to use stable addresses for link local messages and run IPsec tunnel mode to autoconfigure the temporary identifiers, and then use those for the end-to-end traffic.

7. Security Implications of Link Local Messages

In this chapter we discuss the security implications of either securing or not securing the link local messages. In this analysis we will make a few alternative assumptions. The first assumption is that all subsequent communications at a higher level are secured using e.g. IPsec and IKE. It can be argued that making this assumption is reasonable since higher level traffic is already very well secured using existing mechanisms, and if security is of interest, these mechanisms should be used. Note that this assumption has a number of consequences. For instance, being able to redirect a message flow to an attacker doesn’t really gain any information about the flow.

In practice we can not always have secure higher layer communications. First of all, because there is always some traffic that goes to public servers e.g. on the Internet and it is not likely that we will have the trust and key infrastructure necessary to be able to communicate with everyone securely. Secondly, even for other communications it may be that upper layer security does not exist for reasons of missing support in the involved nodes, lack of time to set these up properly, or performance. Therefore, it also also necessary to study the security implications of the link local messages in the case of insecure higher layers.

The second set of alternative assumptions is dividing the tools the attacker has available in two categories. The weak attacker can do nothing more than to inject forged messages. The strong attacker can also block messages, or the whole link. It is interesting to consider these two as separate, because strong measures such as jamming WLAN reception are likely to be detected and less likely to be available to large numbers of attackers, whereas the weak measures are easily available to anyone with standard hardware and software.

We will use the following attacks to classify the dangers:

- Denial-of-Service, e.g. being able to block all traffic to and from a node.

- Impersonation, i.e. the ability of the attacker to claim that it is the node at a particular IP address and not the real node.

- Spying i.e. being able to read packets in transit. In general, spying is always possible for the local traffic unless higher layer security is in place.
- Man-in-the-middle i.e. being able to read and modify packets in transit. In general, this is always possible on a local link for a strong attacker if there is no higher layer security.

- Identity selection i.e. being able to have an effect to the IP address selection of a node.

### 7.1. Router Discovery

First we deal with the secure higher layer assumption. A weak attacker can make the network believe there are additional addresses and prefixes on the link, or that the route towards the Internet is on the other router than it really is. However, since hosts are required to detect when communication to a destination appears to be failing [1] they should be able to find the real router sooner or later. Also, additional address prefixes shouldn’t cause any problems either, except in the case the attacker makes the host believe that a destination somewhere else is actually on the link. In this special case the result is a Denial-of-Service attack.

Strong attackers can also hide prefixes and addresses and routers that really exist on the link. As such, they can perform various kinds of Denial-of-Service attacks.

In the case of insecure higher layers, even weak attackers can read and modify traffic contents towards remote destinations, or impersonate as something they are not. For instance, if the weak attacker has control over a node outside the local link, he can tunnel all traffic he receives from himself to this outside node, resulting in a Man-in-the-Middle attack. The situation may continue for an extended period of time, not allowing the nodes to find out the "real" router but diverting all traffic through the attacker. A strong attacker can perform impersonation on the behalf of any node on the link by hiding the router advertisement with the correct prefix, and claiming to be a router towards the impersonated node. Reading and modification of local traffic is also possible using only router discovery messages through sending different router discovery messages to different nodes on the same network.

### 7.2. Address Resolution

In the secure higher layer situation, weak attackers can cause a Denial-of-Service by making other nodes on the link believe that the node is in some other address than it really is on.

Strong attackers can also prevent correct address resolution messages from being read by others, but the result of this is no worse than the Denial-of-Service already caused by the weak attacker.

In the case of no security at higher layers, both weak and strong attackers can read traffic destined anywhere as well as impersonate other nodes and perform Man-in-the-Middle attacks. Even weak attackers can fake address resolution messages by sending forged messages right after seeing the correct ones.
7.3. Duplicate Address Detection

Regardless of the higher layer security situation, a weak attacker can easily forge messages that cause other nodes on the link to believe their addresses are duplicates. This makes it impossible for them to communicate using these addresses, causing an all-out Denial-of-Service situation.

Strong attackers can cause nodes to believe there is no duplicates when there is one in fact. This will cause several problems later in the communication, in practice leading to a Denial-of-Service situation for all or most communication from the nodes. On the other hand, if there really was a duplicate address situation they would not have been able to communicate in the first place. Therefore, the strong attacker can’t really cause any more damage than the weak one.

When statefull address autoconfiguration is used, both weak and strong attackers can use the duplicate address detection procedure to artificially cause given addresses to fail; it may be possible to force a particular address to be selected by repeatedly blocking the use of certain addresses.

7.4. Address Autoconfiguration

All attacks against the duplicate address detection and router discovery apply also here.

7.5. Neighbour Reachability Detection

The weak attacker can only make a node believe that another node is up when in reality it isn’t. This can be done by sending a forged answer to a reachability detection message. This causes a weak form of a Denial-of-Service attack; the communications would not have succeeded anyway but now one node thinks that it can communicate with the other node. Depending on what kind of communications is in question, this may imply either nothing or delay the attempt of the node to switch to an alternative communications partner.

A strong attacker can also cause the reverse, i.e. make a node believe that another node is down. This results in a full Denial-of-Service attack.

7.6. Redirect

At worst in the secure higher layer case, the attacker succeeds in a Denial-of-Service attack by either constructing a forged Redirect message or blocking a real Redirect message. The latter action is only available to a strong attacker.

In the case of insecure higher layer, any attacker can now reroute non-local traffic to itself, presenting additional opportunities for spying, impersonation and Man-in-the-Middle attacks.
### 7.7. Summary

The following table summarizes the security effects of the link local control functions under different assumption. In the table we have listed the additional dangers resulting e.g. from an insecure address resolution function. The table does not include dangers already inherent in the network otherwise, such as the obvious ability to read all traffic given insecure higher layer protocols.

<table>
<thead>
<tr>
<th>Control Function</th>
<th>Secure higher layer</th>
<th>Insecure higher layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weak Attacker</td>
<td>Strong Attacker</td>
</tr>
<tr>
<td></td>
<td>DoS(R)</td>
<td>DoS</td>
</tr>
<tr>
<td>Router Discovery</td>
<td>Imp(R), MitM(R)</td>
<td>Imp, MitM</td>
</tr>
<tr>
<td>Address Resolution</td>
<td>DoS</td>
<td>DoS, Spy, Imp, MitM</td>
</tr>
<tr>
<td>Duplicate Address Detection</td>
<td>IDSel(Sf)</td>
<td>IDSel(Sf)</td>
</tr>
<tr>
<td>Address Autoconfiguration</td>
<td>DoS, IDSel(Sf)</td>
<td>DoS, IDSel(Sf)</td>
</tr>
<tr>
<td>Neighbour Reachability Detection</td>
<td>DoS(Pd)</td>
<td>DoS(Pd)</td>
</tr>
<tr>
<td>Redirect</td>
<td>DoS(R)</td>
<td>DoS(R), Spy(R), Imp(R), MitM(R)</td>
</tr>
</tbody>
</table>

**Notation:**

- **DoS** = Denial-of-Service
- **Spy** = Listening in
- **Imp** = Impersonation
- **MitM** = Man-in-the-Middle
- **IDSel** = Identity selection (forcing a particular IP address on a host)
- **(R)** = This attack is only possible for traffic destined to hosts outside the local link
- **(Sf)** = This attack is possible only with statefull address autoconfiguration
- **(Pd)** = Weak form; the attack only succeeds in hiding the fact that the other peer is down
These results can be compared to the base situation for payload packets. In there, weak attackers cannot perform any attacks if there is security at higher layers, but strong attackers succeed in Denial-of-Service. If there is no security at higher layers, both weak and strong attackers also succeed in Denial-of-Service and can perform Spying, Impersonation and Man-in-the-Middle attacks. In conclusion, the additional dangers resulting from the insecure control traffic are the following:

- Denial-of-Service for weak attackers under higher layer security
- Man-in-the-Middle, Spying, and Impersonation for all attackers under no higher layer security.
- Identity selection in all situations, if statefull address autoconfiguration is used.

8. Approaches for Reducing Configuration Effort

If security implications of leaving ICMPv6 messages are considered unacceptable, one may question the feasibility of configuring very large networks with manual SAs. There are possible remedies for this situation. These include the following:

(1) Management tools exist at a higher level to create SAs for the link local traffic in an easy manner for the user, but several SAs would still be used on the wire. This has the effect of removing the configuration problems for the user while still being able to use existing implementations as is. However, resources such as memory must still be reserved for the SAs.

(2) A well-known SPI number could be reserved from the IANA-administered range (0-255) to signify link local message protection. In conjunction with this, the rules for processing incoming IPsec packets would have to be changed in order to ignore the destination address for this special SPI. This has implications to existing implementations, and potentially even hardware-based packet matching and/or SA search.

A fixed set of SA parameters must be used throughout the local network. Obviously, replay protection can’t be employed as several nodes use the same SAs. It might be possible to use more than SPI number to enable the simultaneous use of several SAs during a manual key change period (such changes would typically be long-lasting since e.g. laptops might easily be away from their home LAN for weeks).

This approach has the advantage that it would enable the use of the address privacy extension [9] as the SAs could be configured without knowledge of the actual physical addresses and/or randomly generated temporary addresses.

(3) In conjunction with the above, an automatic change procedure could be designed to force the nodes to regularly update the local SA from a central server. One of the routers on the link could act as the
server. A set of SPIs would be required in order to maintain at least two SAs while a key change is in progress.

One way to provide such an automatic procedure is to use unprotected communications to find the local DHCP server, establish an IKE connection to it, and then request for ‘local SA data’ from the server through the created IPsec SA. This approach requires the server to be able to negotiate IPsec SAs with clients having the undefined address. This seems to be possible, given that DHCP servers also are able to respond to the undefined address through replying to the physical address. The use of IKE in this context would be beneficial, since it contains facilities for certificate-based authentication, making the setting up of a set of nodes for a network easier. Separate shared secrets for each node do not scale well, and the use of group shared key has both security drawbacks and is hard to handle in situations where some of the nodes are away for long times.

Since at boot stage the finding of the server would have to be done through unprotected communications, the clients should try to establish IKE and DHCP with all responding routers, in case attackers on the link try to perform a Denial-of-Service attack by responding with fake router addresses.

(4) A unicast key management protocol and a group key management protocol would be used to create SAs for the ICMPv6 traffic. It isn’t clear, however, if this is feasible. Reference [4] already describes how IKE runs into chicken and egg problems in this area; and this is not simply a problem of IKE but rather a fundamental limitation in communications architecture. Likewise, it is likely that multicast key management schemes such as those worked by the multicast key management WG in the IETF will run into similar problems.

(5) An ICMPv6-level key management protocol would be developed. Such a protocol would run as a part of the ICMPv6 operations using e.g. physical link addresses to exchange packets, and would not require passing through the rest of the ICMPv6 processes before any contact could be established. This seems to be a possible approach and ideas along this approach have been presented [8], though most probably the result would quite complex. The protocol would have to handle both unicast and multicast key management.

9. Conclusions

When payload traffic packets are protected e.g. using IPsec, the most serious attacks ‘‘weak attackers’’ are able to mount on IPv6 link local control message result in Denial-of-Service. When these messages, too, are protected with IPsec in the manner described in this draft, these attacks can be prevented. Against a ‘‘strong attacker’’, Denial-of-Service can not be prevented, as he can for instance jam the whole link. However, such attacks are perhaps a little bit easier to detect. Without security for payload packets, the situation is more serious and even weak attackers can perform various kinds of attacks, including Impersonation and Spying. This is particularly dangerous as
it is likely that at least some traffic on the Internet will remain unprotected well into the future.

The Denial-of-Service and other attacks can be solved with the use of IPsec on the link local messages. In order to use IPsec, the SAs must be configured manually as described in this draft. A fair amount of configuration work is involved. If smaller amount of work is desired some of the advanced schemes listed in this draft could be employed, though some schemes would have an effect to existing IPsec implementations. In view of the relatively small security impacts of using unprotected link local messages, it is perhaps best to use the most simple methods. However, scalable local link key management may require the use of certificate based authentication.

10. References


11. Author’s Address

Jari Arkko
Oy LM Ericsson Ab
02420 Jorvas
Finland
E-mail: Jari.Arkko@ericsson.com
Tero Kivinen
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 HELSINKI
Finland
E-mail: kivinen@ssh.fi

Pekka Nikander
Oy LM Ericsson Ab
02420 Jorvas
Finland
E-mail: Pekka.Nikander@nomadiclab.com

Markku Rossi
SSH Communications Security Corp
Fredrikinkatu 42
FIN-00100 HELSINKI
Finland
E-mail: mtr@ssh.fi