DHC
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2007

J. Brzozowski
Comcast Cable
K. Kinnear
B. Volz
S. Zeng
Cisco Systems, Inc.
June 15, 2007

DHCPv6 Leasequery
<draft-ietf-dhc-dhcpv6-leasequery-01.txt>

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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 obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on December 17, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document specifies a leasequery exchange for the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) which can be used to obtain lease information about DHCPv6 clients from a DHCPv6 server. This document specifies the scope of data that can be retrieved as well as both DHCPv6 leasequery requestor and server behavior. This document

extends DHCPv6.

Table of Contents

1. Introduction

The DHCPv6 [2] protocol specifies a mechanism for the assignment of both IPv6 address and configuration information to IPv6 nodes. IPv6 Prefix Options for DHCPv6 [4] specifies a mechanism for the automated delegation of IPv6 prefixes and related options. Similar to DHCPv4 [5], DHCPv6 servers maintain authoritative information related to its operations including but not limited to lease information for IPv6 addresses and delegated prefixes.

The requirement exists in various types of IPv6 deployments, particularly those of a broadband variety, to leverage DHCPv6 [2] for retrieving data related to the operation of DHCPv6 servers programmatically. In particular it is desirable to be able to extract lease information about IPv6 addresses and delegated prefixes assigned using DHCPv6 [2] [4]. Specific examples where this information has illustrated value are in broadband networks to facilitate access control by edge devices. This capability to programmatically extract lease data from the DHCPv6 server is called leasequery.

The leasequery capability described in this document parallels the DHCPv4 leasequery capability documented in [3]. As such, it shares the basic motivations, background, design goals and constraints as described in [3]. Differences are due to the differences between IPv4 and IPv6 and by extension, DHCPv4 and DHCPv6. For example, Neighbor Discovery [7] is used in IPv6 instead of ARP [8] (section 4.1 of [3]) and DOCSIS 3.0 [11] defines IPv6 support for cable modem environments.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

DHCPv6 terminology is defined in [2]. Terminology specific to DHCPv6 leasequery can be found below:

access concentrator
An access concentrator is a router or switch at the broadband access provider's edge of a public broadband access network. This document assumes that the access concentrator includes the DHCPv6 relay agent functionality.

client(s)
The nodes that have one or more bindings with a DHCPv6 server. This does not refer to the node issuing the LEASEQUERY unless it itself has one or more bindings with a DHCPv6 server.

gleaning
Gleaning is the extraction of location information from DHCPv6 messages, as the messages are forwarded by the DHCP relay agent function.

location information
Location information is information needed by the access concentrator to forward traffic to a broadband-accessible host. This information includes knowledge of the host hardware address, the port or virtual circuit that leads to the host, and/or the hardware address of the intervening subscriber modem.

requestor
The node that sends LEASEQUERY messages to one or more servers to retrieve information on the bindings for a client.

3. Protocol Overview

The focus of this document is to extend the DHCPv6 protocol to allow processes and devices that wish to access information from a DHCPv6 server to do so in a lightweight and convenient manner. It is especially appropriate for processes and devices that already interpret DHCPv6 messages.

The LEASEQUERY message is a query message only and does not affect the state of the IPv6 address or prefix, or the binding information associated with it.

One important motivating example is that the LEASEQUERY message allows access concentrators to query DHCP servers to obtain location information of broadband access network devices. This is described

in section 1 of [3] for IPv4.

3.1. On-Demand Query

The on-demand leasequery capability allows requesting just the information necessary to satisfy an immediate need. If the requestor is an access concentrator, then the immediate need will typically be that it has received an IPv6 packet and it needs to refresh its information concerning the DHCPv6 client to which that an IPv6 address is currently leased. In this case, the request will be by Address. This fits clearly into the single request/response cycle common to other DHCPv6 message exchanges.

However, this approach has limitations when used with prefix delegation [4] as no traffic may arrive because the access concentrator is unable to inject the appropriate routing information into the routing infrastructure, such as after a reboot. This approach does work if the access concentrator is configured to inject routing information for a prefix which aggregates potentially delegated prefixes. Or, if the access concentrator and requesting router use a routing protocol; as then the requesting router can trigger the access concentrator to request information from a DHCPv6 server and inject appropriate routing information into the routing infrastructure.

3.2. Anticipatory Query

A second approach for requesting information from a DHCPv6 server would be to use a leasequery-like capability to rebuild an internal data store containing information available from a DHCPv6 server. The rebuilding of the data store in this approach can take place as soon as possible after the need to rebuild it is discovered (such as on booting), and doesn't wait on the receipt of specific packets to trigger a piecemeal database update (as is the case for on-demand leasequery). This approach would also remove the limitation discussed above for prefix delegation.

This anticipatory query is not specified in this document and is an area of future work.

3.3. Query Types

Leasequery provides for the following queries:

Query by IPv6 address - This query allows a requestor to request from a server the bindings for a client that either is bound to the address or has been delegated the prefix that contains the address.

Query by Client Identifier (DUID) - This query allows a requestor to request from a server the bindings for a specific client on a specific link or a list of the links on which the client has one or more bindings.

4. Protocol Details

4.1. Message and Option Definitions

4.1.1. Messages

The LEASEQUERY and LEASEQUERY-REPLY messages use the Client/Server message formats described in [2], section 6. Two new message codes are defined:

LEASEQUERY (TBD) - A requestor sends a LEASEQUERY message to any available server to obtain information on a client's or clients' leases. The options in an OPTION_LQ_QUERY determine the query.

LEASEQUERY-REPLY (TBD) - A server sends a LEASEQUERY-REPLY message containing client data in response to a LEASEQUERY message.

4.1.2. Options

4.1.2.1. Query Option

The Query option is used only in a LEASEQUERY message and identifies the query being performed. The option includes the query type, link-address (or 0::0), and option(s) to provide data needed for the query.

The format of the Query option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_LQ_QUERY        |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   query-type  |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
|                         link-address                          |
|                                                               |
|               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                                               .
+-+-+-+-+-+-+-+-+                                               .
.                         query-options                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code   OPTION_LQ_QUERY (TBD)

option-len   17 + length of query-options field.

link-address   A global address that will be used by the server to identify the link to which the query applies, or 0::0 if unspecified.

query-type   the query requested (see below).

query-options   the options related to the query.

The query-type and required query-options are:

QUERY_BY_ADDRESS (1) - The query-options MUST contain an OPTION_IAADDR option [2]. The link-address field, if not 0::0, specifies an address for the link on which the client is located if the address in the OPTION_IAADDR option is of insufficient scope. Only the information for the client that has a lease for the specified address or was delegated a prefix that contains the specified address is returned (if available).

QUERY_BY_CLIENTID (2) -   The query-options MUST contain an OPTION_CLIENTID option [2]. The link-address field, if not 0::0, specifies an address for the link on which the client is located. If the link-address field is 0::0, the server SHOULD search all of its links of the client.

The query-options MAY also include an OPTION_ORO option [2] to indicate the options for each client that the requestor would like the server to return. Note that this OPTION_ORO is distinct and separate from an OPTION_ORO that may be in the requestor's LEASEQUERY message.

If a server receives an OPTION_LQ_QUERY with a query-type it does not support, the server SHOULD return an UnknownQueryType status-code. If a server receives a supported query-type but the query-options is missing a required option, the server SHOULD return a MalformedQuery status-code.

4.1.2.2. Client Data Option

The Client Data option is used to encapsulate the data for a single client on a single link in a LEASEQUERY-REPLY message.

The format of the Client Data option is shown below:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |       OPTION_CLIENT_DATA      |         option-len            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                               .
 .                        client-options                         .
 .                                                               .
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code OPTION_CLIENT_DATA (TBD)

option-len length, in octets, of the encapsulated client-options field.

client-options the options associated with this client.

The encapsulated client-options include the OPTION_CLIENTID, OPTION_IAADDR, OPTION_IAPREFIX, and OPTION_CLT_TIME options and other options specific to the client and requested by the requestor in the OPTION_ORO in the OPTION_LQ_QUERY's query-options. The server MUST return all of the client's statefully assigned addresses and delegated prefixes, with a non-zero valid lifetime, on the link.

4.1.2.3. Client Last Transaction Time Option

The Client Last Transaction Time option is encapsulated in an OPTION_CLIENT_DATA and identifies how long ago the server last communicated with the client, in seconds.

The format of the Client Last Transaction Time option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_CLT_TIME        |          option-len           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 client-last-transaction-time                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code OPTION_CLT_TIME (TBD)

option-len 4

client-last-transaction-time
the number of seconds since the server last communicated with the client (on that link).

The client-last-transaction-time is a positive value and reflects the number of seconds since the server last communicated with the client (on that link).

4.1.2.4. Relay Data

The Relay Data option is used only in a LEASEQUERY-REPLY message and provides the relay agent information used when the client last communicated with the server.

The format of the Relay Data option is shown below:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     OPTION_LQ_RELAY_DATA      |         option-len            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                  peer-address (IPv6 address)                  |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       DHCP-relay-message                      |
    .                                                               .
    .                                                               .
    .                                                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code   OPTION_LQ_RELAY_DATA (TBD)

option-len   16 + length of DHCP-relay-message.

peer-address   The address of the relay agent from which the relayed message was received by the server.

DHCP-relay-message
The last complete relayed message excluding the client's message OPTION_RELAY_MSG received by the server.

This option is used by the server to return full relay agent information for a client. It MUST NOT be returned if the server does not have such information, either because the client communicated directly (without relay agent) with the server or if the server did not retain such information.

If returned, the DHCP-relay-message MUST contain a valid (perhaps multi-hop) RELAY-FORW message as most recently received by the server for the client. However, the (inner most) OPTION_RELAY_MSG option containing the client's message MUST have been removed.

This option SHOULD only be returned if requested by the OPTION_ORO of the OPTION_LQ_QUERY.

4.1.2.5. Client Link Option

The Client Link option is used only in a LEASEQUERY-REPLY message and identifies the links on which the client has one or more bindings. It is used in reply to a query when no link-address was specified and the client is found to be on more than one link.

The format of the Client Link option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPTION_LQ_CLIENT_LINK     |          option-len           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  link-address (IPv6 address)                  |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  link-address (IPv6 address)                  |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code OPTION_LQ_CLIENT_LINKS (TBD)

option-len Length of the list of links in octets; must be a multiple of 16.

link-address A global address used by the server to identify the link on which the client is located.

A server may respond to a query by client-id, where the 0::0 link-address was specified, with this option if the client is found to be on multiple links. The requestor may then repeat the query once for each link-address returned in the list, specifying the returned link-address. If the client is on a single link, the server SHOULD return the client's data in an OPTION_CLIENT_DATA option.

4.1.3. Status Codes

The following new status codes are defined:

UnknownQueryType (TBD) - The query-type is unknown to or not supported by the server.

MalformedQuery (TBD) - The query is not valid, for example a required query-option is missing from the OPTION_LQ_QUERY.

NotConfigured (TBD) - The server does not have the target address or link in its configuration.

NotAllowed (TBD) - The server does not allow the requestor to issue this LEASEQUERY.

4.1.4. Transmission and Retransmission Parameters

This section presents a table of values used to describe the message transmission behavior for leasequery.

Parameter	Default	Description
LQ_TIMEOUT	1 sec	Initial LEASEQUERY timeout
LQ_MAX_RT	10 secs	Max LEASEQUERY timeout value
LQ_MAX_RC	5	Max LEASEQUERY retry attempts

4.2. Message Validation

4.2.1. LEASEQUERY

Requestors and clients MUST discard any received LEASEQUERY messages.

Servers MUST discard any received LEASEQUERY messages that meet any of the following conditions:

- o the message does not include an OPTION_CLIENTID option.
- o the message includes an OPTION_SERVERID option but the contents of the OPTION_SERVERID option does not match the server's identifier.
- o the message does not include an OPTION_LQ_QUERY option.

4.2.2. LEASEQUERY-REPLY

Requestors MUST discard any received LEASEQUERY-REPLY messages that meet any of the following conditions:

- o the message does not include an OPTION_SERVERID option.
- o the message does not include an OPTION_CLIENTID option or the contents of the OPTION_CLIENTID option do not match the DUID of the requestor.
- o the "transaction-id" field in the message does not match the value used in the original message.

Servers and Relay Agents (on the server port, 547 [2]) MUST discard any received LEASEQUERY-REPLY messages.

4.3. DHCPv6 Leasequery Requestor Behavior

This section describes how a requestor initiates lease data retrieval from DHCPv6 servers.

4.3.1. Creation of LEASEQUERY

The requestor sets the "msg-type" field to LEASEQUERY. The requestor generates a transaction ID and inserts this value in the "transaction-id" field.

The requestor MUST include an OPTION_CLIENTID option to identify itself to the server.

The requestor MUST include an OPTION_LQ_QUERY option and set the query-type, link-address, and query-options as appropriate to the query-type (Section 4.1.2.1).

The requestor SHOULD include an OPTION_SERVERID if it is not unicasting the LEASEQUERY yet only wants a response from a specific server.

4.3.2. Transmission of LEASEQUERY

The requestor MAY be configured to use a list of destination addresses, which MAY include unicast addresses, the All_DHCP_Servers multicast address, or other addresses selected by the network administrator. If the requestor has not been explicitly configured, it MAY use the All_DHCP_Servers multicast address as the default.

The requestor SHOULD send LEASEQUERY to one or more DHCPv6 servers which are known to possess authoritative information concerning the query target.

In the absence of information concerning which DHCPv6 servers might possess authoritative information on the query target, the requestor SHOULD send LEASEQUERY to all DHCPv6 servers that the requestor knows about or is configured with. For example, the requestor MAY send LEASEQUERY to the All_DHCP_Servers multicast address.

The requestor transmits LEASEQUERY messages according to section 14 of [2], using the following parameters:

```
IRT    LQ_TIMEOUT
MRT    LQ_MAX_RT
MRC    LQ_MAX_RC
MRD    0
```

If the message exchange fails, the requestor takes an action based on the requestor's local policy. Examples of actions the requestor might take include:

- o Select another server from a list of servers known to the requestor.
- o Send to multiple servers by multicasting to the All_DHCP_Servers address.
- o Terminate the request.

4.3.3. Receipt of LEASEQUERY-REPLY

A successful LEASEQUERY-REPLY is one without an OPTION_STATUS_CODE option (or an OPTION_STATUS_CODE option with a success code). There are three variants:

1. If the server had bindings for the requested client, the message includes an OPTION_CLIENT_DATA option and the requestor extracts the client data from the LEASEQUERY-REPLY and updates its binding information database. If the OPTION_CLIENT_DATA contains no OPTION_CLT_TIME, the requestor SHOULD silently discard the OPTION_CLIENT_DATA option.
2. If the server found bindings for the client on multiple links, the message includes an OPTION_CLIENT_LINK option. The requestor will need to reissue LEASEQUERY messages using each of the returned link-addresses to obtain the client's bindings.
3. If the server had no bindings for the client, neither the OPTION_CLIENT_DATA nor OPTION_CLIENT_LINK option will be present.

An unsuccessful LEASEQUERY-REPLY is one that has an OPTION_STATUS_CODE with an error code. Depending on the status code, the requestor may try a different server (such as for NotAllowed, NotConfigured, and UnknownQueryType), try a different or corrected query (such as for UnknownQueryType and MalformedQuery), or terminate the query.

4.3.4. Handling DHCPv6 Client Data from Multiple Sources

A requestor may receive lease data on the same client from the same DHCPv6 server in response to different types of LEASEQUERY. If a LEASEQUERY is sent to multiple servers, the requestor may receive from several servers lease data on the same DHCPv6 client. This section describes how the requestor handles multiple lease data sources on the same DHCPv6 client from the same server or different

servers.

The client data from the different sources may be disjoint or overlapping. The disjoint and overlapping relationship can happen between data from the same server or different servers.

If client data from two sources on the same client are of different types or values, then the data are disjoint. An example of data of different types is when a requestor receives an IPv6 address lease from one server and a prefix lease from another server, both assigned to the same client. An example of different values (but the same type) is when a requestor receives two IPv6 address leases from two different servers, both assigned to the same client, but the leases are on two different IPv6 addresses. If the requestor receives disjoint client data from different sources, it SHOULD merge them.

If client data from two sources on the same client are of the same type and value, then the data are overlapping. An example of overlapping data is when a requestor receives a lease on the same IPv6 address from two different servers. Overlapping client data are also called conflicting data.

The requestor SHOULD use the OPTION_CLT_TIME to resolve data conflicts originated from different servers, and SHOULD accept data with most recent OPTION_CLT_TIME.

4.4. DHCPv6 Leasequery Server Behavior

A DHCPv6 server sends LEASEQUERY-REPLY messages in response to valid LEASEQUERY messages it receives to return the statefully assigned addresses, delegated prefixes, and other information about that match the query.

4.4.1. Receipt of LEASEQUERY Messages

Upon receipt of a valid LEASEQUERY message, the DHCPv6 server locates the requested client, collects data on the client, and constructs and returns a LEASEQUERY-REPLY. A LEASEQUERY message can not be used to assign, release, or otherwise modify bindings or other configuration information.

The server constructs a LEASEQUERY-REPLY message by setting the "msg-type" field to LEASEQUERY-REPLY, and copying the transaction ID from the LEASEQUERY message into the transaction-id field.

If the query-type in the OPTION_LQ_QUERY option is not a known or supported value, the server adds an OPTION_STATUS_CODE option with the UnknownQueryType status code and sends the LEASEQUERY-REPLY to

the requestor. If the query-options do not contain the required options for the query-type, the server adds an OPTION_STATUS_CODE option with the MalformedQuery status code and sends the LEASEQUERY-REPLY to the client.

A server may also restrict LEASEQUERY messages, or query-types, to certain requestors. In this case, the server MAY discard the LEASEQUERY message or MAY add an OPTION_STATUS_CODE option with the NotAllowed status code and send the LEASEQUERY-REPLY to the requestor.

If the OPTION_LQ_QUERY specified a non-zero link-address, the server MUST use the link-address to find the appropriate link for the client. For a QUERY_BY_ADDRESS, if the 0::0 link-address was specified, the server uses the address from the OPTION_IAADDR option to find the appropriate link for the client. In either of these cases, if the server is unable to find the link, it SHOULD return an OPTION_STATUS_CODE option with the NotConfigured status and send the LEASEQUERY-REPLY to the requestor.

For a QUERY_BY_CLIENTID, if a 0::0 link-address was specified, the server MUST search all of its links for the client. If the client is only found on a single link, the server SHOULD return that client's data in an OPTION_CLIENT_DATA option. If the client is found on more than a single link, the server MUST return the list of links in the OPTION_CLIENT_LINK option; the server MUST NOT return any client data.

Otherwise, the server uses the data in the OPTION_LQ_QUERY to initiate the query. The result of the query will be zero or one client. This will result in zero or one OPTION_CLIENT_DATA option being added to the LEASEQUERY-REPLY.

4.4.2. Constructing the Client's OPTION_CLIENT_DATA

An OPTION_CLIENT_DATA option in a LEASEQUERY-REPLY message MUST minimally contain the following options:

1. OPTION_CLIENTID
2. OPTION_IAADDR and/or OPTION_IAPREFIX
3. OPTION_CLT_TIME

Depending on the bindings the client has on a link, either OPTION_IAADDR options, OPTION_IAPREFIX options, or both may be present.

The OPTION_CLIENT_DATA SHOULD include options requested in the OPTION_ORO of the OPTION_LQ_QUERY option in the LEASEQUERY message and that are acceptable to return based on the list of "sensitive

options", discussed below.

DHCPv6 servers SHOULD be configurable with a list of "sensitive options" that must not be returned to the requestor when specified in the OPTION_ORO of the OPTION_LQ_QUERY option in the LEASEQUERY message. Any option on this list MUST NOT be returned to a requestor, even if requested by that requestor.

4.4.3. Transmission of LEASEQUERY-REPLY Messages

The server sends the LEASEQUERY-REPLY message as described in the "Transmission of Reply Messages" section of [2].

5. Security Considerations

Access concentrators are expected to be common leasequery requestors. Access concentrators that use DHCPv6 gleaning (i.e., [10]), refreshed with LEASEQUERY messages, will maintain accurate client/binding information. This ensures that the access concentrator can forward data traffic to the intended destination in the broadband access network, can perform IPv6 source address verification of datagrams from the access network, and can encrypt traffic that can only be decrypted by the intended access modem (e.g., [12] and [13]). Thus, the leasequery capability allows an access concentrator to provide considerably enhanced security.

The "Security Considerations" section of [2] details the general threats to DHCPv6, and thus to LEASEQUERY messages. The "Authentication of DHCP Messages" section of [2] describes securing communication between relay agents and servers, as well as clients and servers. If the requestor is an access concentrator, the IPsec [9] based security as described in [2] section 21.1 SHOULD be used. Other types of requestors are essentially DHCPv6 clients. Thus, DHCPv6 authentication, section 21 of [2], is an appropriate mechanism for securing LEASEQUERY and LEASEQUERY-REPLY messages. As the number of leasequery requestors and servers in an administrative domain is relatively small, any shared key distribution issues are minimized.

After implementing the above approaches, the DHCPv6 server should only be communicating with trusted LEASEQUERY requestors, and so security needs should be met.

However, not all traffic originates directly from these trusted requestors. For example, trusted relay agents can relay LEASEQUERY messages from untrusted requestors or elsewhere in the network. This SHOULD be prevented at least at the perimeter relay agents (or on all relay agents unless relayed LEASEQUERY messages are required for some

requestors). DHCPv6 servers MAY be configured to discard relayed LEASEQUERY messages or restrict relay chaining.

DHCPv6 servers SHOULD also provide for the ability to restrict the information returned for a client in a LEASEQUERY-REPLY even to a trusted LEASEQUERY requestor, as described in Section 4.4.2.

Since even trusted access concentrators may generate LEASEQUERY requests as a result of activity external to the access concentrator, access concentrators SHOULD minimize potential denial of service attacks on the DHCPv6 servers by minimizing the generation of LEASEQUERY messages. In particular, the access concentrator SHOULD employ negative caching (i.e., cache the fact that a particular recent query failed to return client data) and address restrictions where possible (i.e., don't send a LEASEQUERY message for addresses outside of the range of the attached broadband access networks). Together, these mechanisms limit the access concentrator to transmitting one LEASEQUERY message (excluding message retries) per legitimate broadband access network address after a reboot event.

Packet flooding denial of service attacks can result in the exhaustion of processing resources, thus preventing the server from serving legitimate and regular DHCPv6 clients as well as legitimate DHCPv6 LEASEQUERY requestors, denying configurations to legitimate DHCPv6 clients as well lease information to legitimate DHCPv6 LEASEQUERY requestors. While these attacks are unlikely when only communicating with trusted LEASEQUERY requestors, the possibility always exists that the trust is misplaced, security techniques are compromised, or even trusted requestors can have bugs in them. Therefore techniques for defending against packet flooding denial of service are always a good idea, and they include good perimeter security as mentioned earlier and rate limiting DHCPv6 traffic by relay agents, other network elements, or the server itself.

One way to attack an access concentrator (as opposed to a DHCPv6 server) as a LEASEQUERY requestor is the establishment of a malicious server with the intent of providing incorrect lease or route information to the access concentrator, thwarting source IPv6 address verification and preventing correct routing. This type of attack can be minimized by using IPsec as described in section 21.1 of [2].

6. IANA Considerations

IANA is requested to assign the following new DHCPv6 Message types in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

LEASEQUERY
LEASEQUERY-REPLY

IANA is requested to assign the following new DHCPv6 Option Codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

OPTION_LQ_QUERY
OPTION_CLIENT_DATA
OPTION_CLT_TIME
OPTION_LQ_RELAY_DATA
OPTION_LQ_CLIENT_LINK

IANA is requested to assign the following new DHCPv6 Status Codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

UnknownQueryType
MalformedQuery
NotConfigured
NotAllowed

IANA is requested to create a new registry for the OPTION_LQ_QUERY option query-type codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters with the following initial assignments:

QUERY_BY_ADDRESS	1
QUERY_BY_CLIENTID	2

New OPTION_LQ_QUERY option query-type codes are assigned through Standards Action, as defined in [6].

7. Acknowledgements

Thanks to Ralph Droms, Richard Johnson, Josh Littlefield, Hemant Singh, Pak Siripunkaw, Markus Stenberg, and Ole Troan for their input, ideas, and review during the production of this document.

8. Modification History

If this section is present in the document when it is submitted for publication, the RFC Editor is requested to remove it.

This document was previously accidentally published under an

incorrect name, draft-ietf-dhc-dhcvp6-leasequery-00 and draft-ietf-dhc-dhcvp6-leasequery-01. The changes between the -00 and -01 version were:

o Added the ability to query by client identifier (DUID), QUERY_BY_CLIENTID. To avoid potentially large messages for clients that are multihomed or mobile, a new option, OPTION_LQ_CLIENT_LINK, to return the list of the links the client is on was added. The requestor then needs to re-query for each link, specifying the link-address in the query to get the client's data.
o Added the ability to return full relay agent details via the OPTION_LQ_RELAY_DATA option.
o And, other minor changes to accommodate the above.

The changes between draft-ietf-dhc-dhcvp6-leasequery-01 and draft-ietf-dhcpv6-leasequery-00 (the corrected document name) were:

o Fixed draft name (had dhcvp6 instead of dhcpv6).
o Removed reference to SRSN I-D and associated text. SRSN is only needed if and when RAAN moves forward (in or close to its current form).
o Added [12] and [13] references that were missing (referenced in Security Considerations section).
o Updated RAAN I-D reference to current version.
o Updated boilerplate and copyright year.

The changes between draft-ietf-dhc-dhcpv6-leaesquery-00 and this version were as a result of the IETF last-call and IESG review:

o Several spelling, typographical, and grammatical corrections were made.
o The Terminology section was expanded to define more of the terms used, using the definitions from RFC 4388 [3].
o The Introduction and Protocol Overview sections were revised to explicitly reference material in RFC 4388 [3] and also indicate some of the key differences.
o The Security Considerations section was reworked.
o The IANA Considerations section now specifies how new query-type codes are assigned - through Standards Action.
o Additional references were added as a result of the above changes.

Thanks to those that pointed out the above issues during the last-call process, including Jari Arkko, Lars Eggert, Sam Hartman, Alfred Hoenes, Tim Polk, and the folks at IANA.

9. References

9.1. Normative References

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

[2] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[3] Woundy, R. and K. Kinnear, "Dynamic Host Configuration Protocol (DHCP) Leasequery", RFC 4388, February 2006.

[4] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

9.2. Informative References

[5] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[6] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[7] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[8] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

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

[10] Droms, R., "DHCPv6 Relay Agent Assignment Notification (RAAN) Option", draft-ietf-dhc-dhcpv6-agentopt-delegate-02 (work in progress), November 2006.

[11] CableLabs, "Data-Over-Cable Service Interface Specifications: DOCSIS 3.0, MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.0-I04-070518", May 2007, available at http://www.cablemodem.com/.

[12] SCTE Data Standards Subcommittee, "Data-Over-Cable Service Interface Specifications: DOCSIS 1.0 Baseline Privacy Interface Specification SCTE 22-2 2002", 2002, available at

http://www.scte.org/standards/.

[13] CableLabs, "Data-Over-Cable Service Interface Specifications: Baseline Privacy Plus Interface Specification CM-SP-BPI+_I12-050812", August 2005, available at http://www.cablemodem.com/.

Authors' Addresses

John Jason Brzozowski
Comcast Cable
1800 Bishops Gate Boulevard
Mt. Laurel, NJ 08054
USA

Phone: +1 856 324 2671
Email: john_brzozowski@cable.comcast.com

Kim Kinnear
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 0000
Email: kkinnear@cisco.com

Bernard Volz
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 0000
Email: volz@cisco.com

Shengyou Zeng
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 0000
Email: szeng@cisco.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).