dhc
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2015

S. Krishnan
Ericsson
T. Mrugalski
ISC
S. Jiang
Huawei Technologies Co., Ltd

October 27, 2014

Privacy considerations for DHCPv6
draft-krishnan-dhc-dhcpv6-privacy-00

Abstract

DHCPv6 is a protocol that is used to provide addressing and configuration information to IPv6 hosts. This document discusses the various identifiers used by DHCPv6 and the potential privacy issues.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

DHCPv6 [RFC3315] is a protocol that is used to provide addressing and configuration information to IPv6 hosts. The DHCPv6 protocol uses several identifiers that could become a source for gleaning additional information about the IPv6 host. This information may include device type, operating system information, location(s) that the device may have previously visited, etc. This document discusses the various identifiers used by DHCPv6 and the potential privacy issues [RFC6973].

Future works may propose protocol changes to fix the privacy issues that have been analyzed in this document. It is out of scope for this document.

Editor notes: for now, the document is mainly considering the privacy of DHCPv6 client. The privacy of DHCPv6 server and relay agent are considered less important because they are open for public services. However, this may be a subject to change if further study shows opposite result.

2. Terminology

This section clarifies the terminology used throughout this document.

Stable identifier - any property disclosed by a DHCPv6 client that does not change over time or changes very infrequently and is unique for said client in a given context. Examples include MAC address, client-id that does not change or a hostname. Stable identifier may or may not be globally unique.

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 [RFC2119]. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

3. Identifiers in DHCPv6

There are several identifiers used in DHCPv6. This section provides an introduction to the various options that will be used further in the document.

3.1. DUID

Each DHCPv6 client and server has a DHCPv6 Unique Identifier (DUID) [RFC3315]. The DUID is designed to be unique across all DHCPv6 clients and servers, and to remain stable after it has been initially generated. The DUID can be of different forms. Commonly used forms are based on the link-layer address of one of the device's network interfaces (with or without a timestamp), on the Universally Unique IDentifier (UUID) [RFC6355]. The default type, recommended by [RFC3315], is DUID-LLT that is based on link-layer address, which is commonly implemented in most popular clients.

It is important to understand DUID lifecycle. Clients and servers are expected to generate their DUID once (during first operation) and store it in a non-volatile storage or use the same deterministic algorithm to generate the same DUID value again. This means that most implementations will use the available link-layer address during its first boot. Even if the administrator enables privacy extensions (see [RFC4941]) and its equivalent for link-layer address randomization, it is likely that those privacy mechanisms were disabled during the first device boot. Hence the original, unobfuscated link-layer address will likely end up being announced as client DUID, even if the link-layer address has changed (or even if being changed on a periodic basis).

3.2. Client ID Option

The Client Identifier Option (OPTION_CLIENTID) [RFC3315] is used to carry the DUID of a DHCPv6 client between a client and a server. There is an analogous Server Identifier Option but it is not as interesting in the privacy context (unless a host can be convinced to start acting as a server). Client ID is an example of DUID. See Section 3.1 for relevant discussion about DUIDs.

3.3. IA_NA, IA_TA, IA_PD, IA Address and IA Prefix Options

The Identity Association for Non-temporary Addresses (IA_NA) option [RFC3315] is used to carry the parameters and any non-temporary addresses associated with the given IA_NA. The Identity Association for Temporary Addresses (IA_TA) option [RFC3315] is analogous to the IA_NA option but for temporary addresses. The IA Address option [RFC3315] is used to specify IPv6 addresses associated with an IA_NA or an IA_TA and is encapsulated within the Options field of such an IA_NA or IA_TA option. The Identity Association for Prefix Delegation (IA_PD) [RFC3633] option is used to carry the prefixes that are assigned to the requesting router. IA Prefix option [RFC3633] is used to specify IPv6 prefixes associated with an IA_PD and is encapsulated within the Options field of such an IA_PD option.

To differentiate between instances of the same type of IA containers, each IA_NA, IA_TA and IA_PD options have an IAID field that is unique for each client/option type pair. It is up to the client to pick unique IAID values. At least one popular implementation uses last four octets of the link-layer address. In most cases, that means that merely two bytes are missing for a full link-layer address reconstruction. However, the first three octets in a typical link-layer address are vendor identifier. That can be determined with high level of certainty using other means, thus allowing full link-layer address discovery.

3.4. Interface ID

A DHCPv6 relay includes the Interface ID [RFC3315] option to identify the interface on which it received the client message that is being relayed.

Although in principle Interface ID can be arbitrarily long with completely random values, it is often a text string that includes the relay agent name followed by interface name. This can be used for fingerprinting the relay or determining client's point of attachment.

3.5. Subscriber ID

A DHCPv6 relay includes a Subscriber ID option [RFC4580] to associate some provider-specific information with clients' DHCPv6 messages that is independent of the physical network configuration.

In many deployments, the relay agent that inserts this option is configured to use client's link-layer address as Subscriber ID.

3.6. Remote ID

A DHCPv6 relay includes a Remote ID option [RFC4649] to identify the remote host end of the circuit.

The remote-id is vendor specific, for which the vendor is indicated in the enterprise-number field. The remote-id field may encode the information that identified the DHCPv6 clients:

- a "caller ID" telephone number for dial-up connection
- a "user name" prompted for by a Remote Access Server
- a remote caller ATM address o a "modem ID" of a cable data modem
- the remote IP address of a point-to-point link

- an interface or port identifier

3.7. Client FQDN Option

The Client Fully Qualified Domain Name (FQDN) option [RFC4704] is used by DHCPv6 clients and servers to exchange information about the client's fully qualified domain name and about who has the responsibility for updating the DNS with the associated AAAA and PTR RRs.

A client can use this option to convey all or part of its domain name to a DHCPv6 server for the IPv6-address-to-FQDN mapping. In most case a client sends its hostname as a hint for the server. The DHCPv6 server MAY be configured to modify the supplied name or to substitute a different name. The server should send its notion of the complete FQDN for the client in the Domain Name field.

3.8. Client Link-layer Address Option

The Client link-layer address option [RFC6939] is used by first-hop DHCPv6 relays to provide the client's link-layer address towards the server.

DHCPv6 relay agents that receive messages originating from clients may include the link-layer source address of the received DHCPv6 message in the Client Link-Layer Address option, in relayed DHCPv6 Relay-Forward messages.

3.9. Option Request Option

DHCPv6 clients include an Option Request option [RFC3315] in DHCPv6 messages to inform the server about options the client wants the server to send to the client.

The content of an Option Request option are the option codes for an option requested by the client. The client may additionally include instances of those options that are identified in the Option Request option, with data values as hints to the server about parameter values the client would like to have returned.

3.10. Vendor Class Option

This Vendor Class option [RFC3315] is used by a DHCPv6 client to identify the vendor that manufactured the hardware on which the client is running.

The information contained in the data area of this option is contained in one or more opaque fields that identify details of the

hardware configuration, for example, the version of the operating system the client is running or the amount of memory installed on the client.

3.11. Civic Location Option

DHCPv6 servers use the Civic Location option [RFC4776] to delivery of location information (the civic and postal addresses) from the DHCPv6 server to the DHCPv6 clients. It may refer to three locations: the location of the DHCPv6 server, the location of the network element believed to be closest to the client, or the location of the client, identified by the "what" element within the option.

3.12. Coordinate-Based Location Option

The GeoLoc options [RFC6225] is used by DHCPv6 server to provide the coordinate- based geographic location information to the DHCPv6 clients. It enable a DHCPv6 client to obtain its location.

After the relevant DHCPv6 exchanges have taken place, the location information is stored on the end device rather than somewhere else, where retrieving it might be difficult in practice.

3.13. Client System Architecture Type Option

The Client System Architecture Type option [RFC5970] is used by DHCPv6 client to send a list of supported architecture types to the DHCPv6 server. It is used to provide configuration information for a node that must be booted using the network rather than from local storage.

4. Existing Mechanisms That Affect Privacy

This section describes available DHCPv6 mechanisms that one can use to protect or enhance one’s privacy.

4.1. Temporary addresses

[RFC3315] defines a mechanism for a client to request temporary addresses. The idea behind temporary addresses is that a client can request a temporary address for a specific purpose, use it, and then never renew it. i.e. let it expire.

There are number of serious issues, both protocolar and implementational, that make them nearly useless for their original goal. First, [RFC3315] does not include T1 and T2 renewal timers in IA_TA (a container for temporary addresses). However, it mentions that temporary addresses can be renewed. Many client implementations

renew those addresses during a renewal procedure initiated by other resources (non-temporary addresses or prefixes), thus forfeiting shortliveness. Second, [RFC4704] allows servers to update DNS for assigned temporary addresses. Publishing client's IPv6 address in DNS that is publicly available is a major privacy breach.

4.2. DNS Updates

DNS Updates [RFC4704] defines a mechanism that allows both clients and server to insert into DNS domain information about clients. Both forward (AAAA) and reverse (PTR) resource records can be updated. This allows other nodes to conveniently refer to a host, despite the fact that its IPv6 address may be changing.

This mechanism exposes two important pieces of information: current address (which can be mapped to current location) and client's hostname. The stable hostname can then by used to correlate the client across different network attachments even when its IPv6 address keeps changing.

4.3. Allocation strategies

A DHCPv6 server running in typical, stateful mode is given a task of managing one or more pools of IPv6 resources (currently non-temporary addresses, temporary addresses and/or prefixes, but more resource types may be defined in the future). When a client requests a resource, server must pick a resource out of configured pool. Depending on the server's implementation, various allocation strategies are possible. Choices in this regard may have privacy implications.

Iterative allocation - a server may choose to allocate addresses one by one. That strategy has the benefit of being very fast, thus can be favored in deployments that prefer performance. However, it makes the resources very predictable. Also, since the resources allocated tend to be clustered at the beginning of available pool, it makes scanning attacks much easier.

Identifier-based allocation - a server may choose to allocate an address that is based on one of available identifiers, e.g. IID or MAC address. This has a property of being convenient for converting IP address to/from other identifiers, especially if the identifier is or contains MAC address. It is also convenient, as returning client is very likely to get the same address, even if the server does not store previous client's address. Those properties are convenient for system administrators, so DHCPv6 server implementors are sometimes requested to implement it. There is at least one implementation that supports it. On the other hand, the downside of such allocation is

that the client now discloses its identifier in its IPv6 address to all services it connects to. That means that correlation of activities over time, location tracking, address scanning and OS/vendor discovery apply.

Hash allocation - it's an extension of identifier based allocation. Instead of using the identifier directly, it is being hashed first. If the hash is implemented correctly, it removes the flaw of disclosing the identifier, a property that eliminates susceptibility to address scanning and OS/vendor discovery. If the hash is poorly implemented (e.g. can be reverted), it introduces no improvement over identifier-based allocation.

Random allocation - a server can pick a resource randomly out of available pool. That strategy works well in scenarios where pool utilization is small, as the likelihood of collision (resulting in the server needing to repeat randomization) is small. With the pool allocation increasing, the collision is disproportionally large, due to birthday paradox. With high pool utilization (e.g. when 90% of available resources being allocated already), the server will use most computational resources to repeatedly pick a random resource, which will degrade its performance. This allocation scheme essentially prevents returning clients from getting the same address or prefix again. On the other hand, it is beneficial from privacy perspective as addresses and prefixes generated that way are not susceptible to correlation attacks, OS/vendor discovery attacks or identity discovery attacks. Note that even though the address or prefix itself may be resilient to a given attack, the client may still be susceptible if additional information is disclosed other way, e.g. client's address can be randomized, but it still can leak its MAC address in client-id option.

Other allocation strategies may be implemented.

5. Attacks

5.1. Device type discovery (fingerprinting)

The type of device used by the client can be guessed by the attacker using the Vendor Class option, the Client Link-layer Address option, and by parsing the Client ID option. All of those options may contain OUI (Organizationally Unique Identifier) that represents the device's vendor. That knowledge can be used for device-specific vulnerability exploitation attacks. See Section 3.4 of [I-D.ietf-6man-ipv6-address-generation-privacy] for a discussion about this type of attack.

5.2. Operating system discovery (fingerprinting)

The operating system running on a client can be guessed using the Vendor Class option, the Client System Architecture Type option, or by using fingerprinting techniques on the combination of options requested using the Option Request option. See Section 3.4 of [I-D.ietf-6man-ipv6-address-generation-privacy] for a discussion about this type of attack.

5.3. Finding location information

The location information can be obtained by the attacker by many means. The most direct way to obtain this information is by looking into a server initiated message that contains the Civic Location or GeoLoc option. It can also be indirectly inferred using the Remote ID Option (e.g. using a telephone number), the Interface ID option (e.g. if an access circuit on an Access Node corresponds to a civic location), or the Subscriber ID Option (if the attacker has access to subscriber info).

5.4. Finding previously visited networks

When DHCPv6 clients connect to a network, they attempt to obtain the same address they had used before they attached to the network. They do this by putting the previously assigned address(es) in the IA Address Option(s) inside the IA_NA, IA_TA. By observing these addresses, an attacker can identify the network the client had previously visited.

5.5. Finding a stable identity

An attacker might use a stable identity gleaned from DHCPv6 messages to correlate activities of a given client on unrelated networks. The Client FQDN option, the Subscriber ID Option and the Client ID options can serve as long lived identifiers of DHCPv6 clients. The Client FQDN option can also provide an identity that can easily be correlated with web server activity logs.

5.6. Pervasive monitoring

This is an enhancement, or a combination of most aforementioned mechanisms. Operator, who controls non-trivial number of access points or network segments, may use obtained information about a single client and observer client’s habits.

5.7. Finding client's IP address or hostname

Many DHCPv6 deployments use DNS Updates [RFC4704] that put client's information (current IP address, client's hostname). Client ID is also disclosed, able it in not easily accessible form (SHA-256 digest of the client-id). Although SHA-256 is irreversible, so DHCPv6 client ID can't be converted back to client-id. However, SHA-256 digest can be used as a unique identifier that is accessible by any host.

5.8. Correlation of activities over time

As with other identifiers, an IPv6 address can be used to correlate the activities of a host for at least as long as the lifetime of the address. If that address was generated from some other, stable identifier and that generation scheme can be deducted by an attacker, the duration of correlation attack extends to that identifier. In many cases, its lifetime is equal to the lifetime of the device itself. See Section 3.1 of [I-D.ietf-6man-ipv6-address-generation-privacy] for detailed discussion.

5.9. Location tracking

If a stable identifier is used for assigning an address and such mapping is discovered by an attacker (e.g. a server that uses IEEE-identifier-based IID to generate IPv6 address), all scenarios discussed in Section 3.2 of [I-D.ietf-6man-ipv6-address-generation-privacy] apply. In particular both passive (a service that the client connects to can log client's address and draw conclusions regarding its location and movement patterns based on prefix it is connecting from) and active (attacker can send ICMPv6 echo requests or other probe packets to networks of suspected client locations).

5.10. Leasequery & bulk leasequery

Attackers may pretend as an access concentrator, either DHCPv6 relay agent or DHCPv6 client, to obtain location information directly from the DHCP server(s) using the DHCPv6 Leasequery [RFC5007] mechanism.

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.

Furthermore, the attackers may use DHCPv6 bulk leasequery [RFC5460] mechanism to obtain bulk information about DHCPv6 bindings, even without knowing the target bindings.

6. Security Considerations

TBD

7. Privacy Considerations

This document at its entirety discusses privacy considerations in DHCPv6. As such, no separate section about this is needed.

8. IANA Considerations

This draft does not request any IANA action.

9. Acknowledgements

The authors would like to thanks the valuable comments made by Stephen Farrell, Ted Lemon, Ines Robles, Russ White, Christian Schaefer and other members of DHC WG.

This document was produced using the xml2rfc tool [RFC2629].

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

10.2. Informative References

[I-D.ietf-6man-ipv6-address-generation-privacy] Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-02 (work in progress), October 2014.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4580] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Subscriber-ID Option", RFC 4580, June 2006.

[RFC4649] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option", RFC 4649, August 2006.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.

[RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", RFC 4776, November 2006.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.

[RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.

[RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, September 2010.

[RFC6225] Polk, J., Linsner, M., Thomson, M., and B. Aboba, "Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information", RFC 6225, July 2011.

[RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355, August 2011.

[RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, May 2013.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

Authors' Addresses

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Tomek Mrugalski
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1345
Email: tomasz.mrugalski@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 BeiQing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: jiangsheng@huawei.com