Network Working Group
Request for Comments: 4701
Category: Standards Track

M. Stapp
Cisco Systems, Inc.
T. Lemon
Nominum, Inc.
A. Gustafsson
Araneus Information Systems Oy

October 2006

A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR)

Status of This Memo

Copyright Notice

Abstract

It is possible for Dynamic Host Configuration Protocol (DHCP) clients to attempt to update the same DNS Fully Qualified Domain Name (FQDN) or to update a DNS FQDN that has been added to the DNS for another purpose as they obtain DHCP leases. Whether the DHCP server or the clients themselves perform the DNS updates, conflicts can arise. To resolve such conflicts, RFC 4703 proposes storing client identifiers in the DNS to unambiguously associate domain names with the DHCP clients to which they refer. This memo defines a distinct Resource Record (RR) type for this purpose for use by DHCP clients and servers: the "DHCID" RR.

Table of Contents

1. Introduction

A set of procedures to allow DHCP [7] [11] clients and servers to automatically update the DNS ([3], [4]) is proposed in [1].

Conflicts can arise if multiple DHCP clients wish to use the same DNS name or a DHCP client attempts to use a name added for another purpose. To resolve such conflicts, [1] proposes storing client identifiers in the DNS to unambiguously associate domain names with the DHCP clients using them. In the interest of clarity, it is preferable for this DHCP information to use a distinct RR type. This memo defines a distinct RR for this purpose for use by DHCP clients or servers: the "DHCID" RR.

In order to obscure potentially sensitive client identifying information, the data stored is the result of a one-way SHA-256 hash computation. The hash includes information from the DHCP client's message as well as the domain name itself, so that the data stored in the DHCID RR will be dependent on both the client identification used in the DHCP protocol interaction and the domain name. This means that the DHCID RDATA will vary if a single client is associated over time with more than one name. This makes it difficult to 'track' a client as it is associated with various domain names.

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 [2].

3. The DHCID RR

The DHCID RR is defined with mnemonic DHCID and type code 49. The DHCID RR is only defined in the IN class. DHCID RRs cause no additional section processing.

3.1. DHCID RDATA Format

The RDATA section of a DHCID RR in transmission contains RDLENGTH octets of binary data. The format of this data and its interpretation by DHCP servers and clients are described below.

DNS software should consider the RDATA section to be opaque. DHCP clients or servers use the DHCID RR to associate a DHCP client's identity with a DNS name, so that multiple DHCP clients and servers may deterministically perform dynamic DNS updates to the same zone. From the updater's perspective, the DHCID resource record RDATA consists of a 2-octet identifier type, in network byte order,

followed by a 1-octet digest type, followed by one or more octets representing the actual identifier:

```
        < 2 octets >    Identifier type code
        < 1 octet >     Digest type code
        < n octets >    Digest (length depends on digest type)
```

3.2. DHCID Presentation Format

In DNS master files, the RDATA is represented as a single block in base-64 encoding identical to that used for representing binary data in [8], Section 3. The data may be divided up into any number of white-space-separated substrings, down to single base-64 digits, which are concatenated to form the complete RDATA. These substrings can span lines using the standard parentheses.

3.3. The DHCID RR Identifier Type Codes

The DHCID RR Identifier Type Code specifies what data from the DHCP client's request was used as input into the hash function. The identifier type codes are defined in a registry maintained by IANA, as specified in Section 7. The initial list of assigned values for the identifier type code and that type's identifier is:

Identifier Type Code	Identifier
0x0000	The 1-octet 'htype' followed by 'hlen' octets of 'chaddr' from a DHCPv4 client's DHCPREQUEST [7].
0x0001	The data octets (i.e., the Type and Client-Identifier fields) from a DHCPv4 client's Client Identifier option [10].
0x0002	The client's DUID (i.e., the data octets of a DHCPv6 client's Client Identifier option [11] or the DUID field from a DHCPv4 client's Client Identifier option [6]).
0x0003 - 0xfffe	Undefined; available to be assigned by IANA.
0xffff	Undefined; RESERVED.

3.4. The DHCID RR Digest Type Code

The DHCID RR Digest Type Code is an identifier for the digest algorithm used. The digest is calculated over an identifier and the canonical FQDN as described in the next section.

The digest type codes are defined in a registry maintained by IANA, as specified in Section 7. The initial list of assigned values for the digest type codes is: value 0 is reserved, and value 1 is SHA-256. Reserving other types requires IETF standards action. Defining new values will also require IETF standards action to document how DNS updaters are to deal with multiple digest types.

3.5. Computation of the RDATA

The DHCID RDATA is formed by concatenating the 2-octet identifier type code with variable-length data.

The RDATA for all type codes other than 0xffff, which is reserved for future expansion, is formed by concatenating the 2-octet identifier type code, the 1-octet digest type code, and the digest value (32 octets for SHA-256).

< identifier-type > < digest-type > < digest >

The input to the digest hash function is defined to be:

digest = SHA-256(< identifier > < FQDN >)

The FQDN is represented in the buffer in the canonical wire format as described in [9], Section 6.2. The identifier type code and the identifier are related as specified in Section 3.3: the identifier type code describes the source of the identifier.

A DHCPv4 updater uses the 0x0002 type code if a Client Identifier option is present in the DHCPv4 messages and it is encoded as specified in [6]. Otherwise, the updater uses 0x0001 if a Client Identifier option is present, and 0x0000 if not.

A DHCPv6 updater always uses the 0x0002 type code.

3.5.1. Using the Client's DUID

When the updater is using the Client's DUID (either from a DHCPv6 Client Identifier option or from a portion of the DHCPv4 Client Identifier option encoded as specified in [6]), the first two octets of the DHCID RR MUST be 0x0002, in network byte order. The third octet is the digest type code (1 for SHA-256). The rest of the DHCID RR MUST contain the results of computing the SHA-256 hash across the octets of the DUID followed by the FQDN.

3.5.2. Using the Client Identifier Option

When the updater is using the DHCPv4 Client Identifier option sent by the client in its DHCPREQUEST message, the first two octets of the DHCID RR MUST be 0x0001, in network byte order. The third octet is the digest type code (1 for SHA-256). The rest of the DHCID RR MUST contain the results of computing the SHA-256 hash across the data octets (i.e., the Type and Client-Identifier fields) of the option, followed by the FQDN.

3.5.3. Using the Client's htype and chaddr

When the updater is using the client's link-layer address as the identifier, the first two octets of the DHCID RDATA MUST be zero. The third octet is the digest type code (1 for SHA-256). To generate the rest of the resource record, the updater computes a one-way hash using the SHA-256 algorithm across a buffer containing the client's network hardware type, link-layer address, and the FQDN data. Specifically, the first octet of the buffer contains the network hardware type as it appeared in the DHCP 'htype' field of the client's DHCPREQUEST message. All of the significant octets of the 'chaddr' field in the client's DHCPREQUEST message follow, in the same order in which the octets appear in the DHCPREQUEST message. The number of significant octets in the 'chaddr' field is specified in the 'hlen' field of the DHCPREQUEST message. The FQDN data, as specified above, follows.

3.6. Examples

3.6.1. Example 1

A DHCP server allocates the IPv6 address 2001:DB8::1234:5678 to a client that included the DHCPv6 client-identifier option data 00:01:00:06:41:2d:f1:66:01:02:03:04:05:06 in its DHCPv6 request. The server updates the name "chi6.example.com" on the client's behalf and uses the DHCP client identifier option data as input in forming a DHCID RR. The DHCID RDATA is formed by setting the two type octets to the value 0x0002, the 1-octet digest type to 1 for SHA-256, and performing a SHA-256 hash computation across a buffer containing the 14 octets from the client-id option and the FQDN (represented as specified in Section 3.5).

```
chi6.example.com.     AAAA    2001:DB8::1234:5678
chi6.example.com.     DHCID   ( AAIBY2/AuCccgoJbsaxcQc9TUapptP69l
                                OjxfNuVAA2kjEA= )
```

If the DHCID RR type is not supported, the RDATA would be encoded [13] as:

```
\# 35 ( 000201636fc0b8271c82825bb1ac5c41cf5351aa69b4febd94e8f17cd
       b95000da48c40 )
```

3.6.2. Example 2

A DHCP server allocates the IPv4 address 192.0.2.2 to a client that included the DHCP client-identifier option data 01:07:08:09:0a:0b:0c in its DHCP request. The server updates the name "chi.example.com" on the client's behalf and uses the DHCP client identifier option data as input in forming a DHCID RR. The DHCID RDATA is formed by setting the two type octets to the value 0x0001, the 1-octet digest type to 1 for SHA-256, and performing a SHA-256 hash computation across a buffer containing the seven octets from the client-id option and the FQDN (represented as specified in Section 3.5).

```
chi.example.com.      A       192.0.2.2
chi.example.com.      DHCID   ( AAEBOSD+XR3Os/0LozeXVqcNc7FwCfQdW
                                L3b/NaiUDlW2No= )
```

If the DHCID RR type is not supported, the RDATA would be encoded [13] as:

```
\# 35 ( 0001013920fe5d1dceb3fd0ba3379756a70d73b17009f41d58bddbfcd
       6a2503956d8da )
```

3.6.3. Example 3

A DHCP server allocating the IPv4 address 192.0.2.3 to a client with the Ethernet MAC address 01:02:03:04:05:06 using domain name "client.example.com" uses the client's link-layer address to identify the client. The DHCID RDATA is composed by setting the two type octets to zero, the 1-octet digest type to 1 for SHA-256, and performing an SHA-256 hash computation across a buffer containing the 1-octet 'htype' value for Ethernet, 0x01, followed by the six octets of the Ethernet MAC address, and the domain name (represented as specified in Section 3.5).

```
client.example.com.   A       192.0.2.3
client.example.com.   DHCID   ( AAABxLmlskllE0MVjd57zHcWmEH3pCQ6V
                                ytcKD//7es/deY= )
```

If the DHCID RR type is not supported, the RDATA would be encoded [13] as:

```
\# 35 ( 000001c4b9a5b249651343158dde7bcc77169841f7a4243a572b5c283
       fffedeb3f75e6 )
```

4. Use of the DHCID RR

This RR MUST NOT be used for any purpose other than that detailed in [1]. Although this RR contains data that is opaque to DNS servers, the data must be consistent across all entities that update and interpret this record. Therefore, new data formats may only be defined through actions of the DHC Working Group, as a result of revising [1].

5. Updater Behavior

The data in the DHCID RR allows updaters to determine whether more than one DHCP client desires to use a particular FQDN. This allows site administrators to establish policy about DNS updates. The DHCID RR does not establish any policy itself.

Updaters use data from a DHCP client's request and the domain name that the client desires to use to compute a client identity hash, and then compare that hash to the data in any DHCID RRs on the name that they wish to associate with the client's IP address. If an updater discovers DHCID RRs whose RDATA does not match the client identity that they have computed, the updater SHOULD conclude that a different client is currently associated with the name in question. The updater SHOULD then proceed according to the site's administrative policy. That policy might dictate that a different name be selected, or it might permit the updater to continue.

6. Security Considerations

The DHCID record as such does not introduce any new security problems into the DNS. In order to obscure the client's identity information, a one-way hash is used. Further, in order to make it difficult to 'track' a client by examining the names associated with a particular hash value, the FQDN is included in the hash computation. Thus, the RDATA is dependent on both the DHCP client identification data and on each FQDN associated with the client.

However, it should be noted that an attacker that has some knowledge, such as of MAC addresses commonly used in DHCP client identification data, may be able to discover the client's DHCP identify by using a brute-force attack. Even without any additional knowledge, the number of unknown bits used in computing the hash is typically only 48 to 80.

Administrators should be wary of permitting unsecured DNS updates to zones, whether or not they are exposed to the global Internet. Both DHCP clients and servers SHOULD use some form of update authentication (e.g., [12]) when performing DNS updates.

7. IANA Considerations

IANA has allocated a DNS RR type number for the DHCID record type.

This specification defines a new number-space for the 2-octet identifier type codes associated with the DHCID RR. IANA has established a registry of the values for this number-space. Three initial values are assigned in Section 3.3, and the value 0xFFFF is reserved for future use. New DHCID RR identifier type codes are assigned through Standards Action, as defined in [5].

This specification defines a new number-space for the 1-octet digest type codes associated with the DHCID RR. IANA has established a registry of the values for this number-space. Two initial values are assigned in Section 3.4. New DHCID RR digest type codes are assigned through Standards Action, as defined in [5].

8. Acknowledgements

Many thanks to Harald Alvestrand, Ralph Droms, Olafur Gudmundsson, Sam Hartman, Josh Littlefield, Pekka Savola, and especially Bernie Volz for their review and suggestions.

9. References

9.1. Normative References

[1] Stapp, M. and B. Volz, "Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients", RFC 4703, October 2006.

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

[3] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[4] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[6] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", RFC 4361, February 2006.

9.2. Informative References

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

[8] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

[9] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[10] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

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

[12] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[13] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.

Authors' Addresses

Mark Stapp
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: 978.936.1535
EMail: mjs@cisco.com

Ted Lemon
Nominum, Inc.
950 Charter St.
Redwood City, CA 94063
USA

EMail: mellon@nominum.com

Andreas Gustafsson
Araneus Information Systems Oy
Ulappakatu 1
02320 Espoo
Finland

EMail: gson@araneus.fi

Full Copyright Statement

Intellectual Property

Acknowledgement

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