DNSEXT Working Group
Internet-Draft
Expires: May 22, 2002

M. Stapp
Cisco Systems, Inc.
T. Lemon
A. Gustafsson
Nominum, Inc.
November 21, 2001

A DNS RR for Encoding DHCP Information (DHCID RR)
<draft-ietf-dnsext-dhcid-rr-04.txt>

Status of this Memo

Copyright Notice

Abstract

It is possible for multiple DHCP clients to attempt to update the same DNS FQDN as they obtain DHCP leases. Whether the DHCP server or the clients themselves perform the DNS updates, conflicts can arise. To resolve such conflicts, "Resolution of DNS Name Conflicts"[1] 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 RR type for this purpose for use by DHCP clients and servers, the "DHCID" RR.

Table of Contents

1. 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 RFC 2119[2].

2. Introduction

A set of procedures to allow DHCP[3] clients and servers to automatically update the DNS (RFC1034[4], RFC1035[5]) is proposed in "Resolution of DNS Name Conflicts"[1].

Conflicts can arise if multiple DHCP clients wish to use the same DNS name. To resolve such conflicts, "Resolution of DNS Name 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 avoid exposing potentially sensitive identifying information, the data stored is the result of a one-way MD5[6] hash computation. The hash includes information from the DHCP client's REQUEST 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.

The MD5 hash algorithm has been shown to be weaker than the SHA-1 algorithm; it could therefore be argued that SHA-1 is a better choice. However, SHA-1 is significantly slower than MD5. A successful attack of MD5's weakness does not reveal the original data that was used to generate the signature, but rather provides a new set of input data that will produce the same signature. Because we are using the MD5 hash to conceal the original data, the fact that an attacker could produce a different plaintext resulting in the same MD5 output is not significant concern.

3. The DHCID RR

The DHCID RR is defined with mnemonic DHCID and type code [TBD]. The DHCID RR is only defined in the IN class. DHCID RRs cause no additional section processing. The DHCID RR is not a singleton type.

3.1 DHCID RDATA format

The RDATA section of a DHCID RR in transmission contains RDLENGTH bytes 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 16-bit identifier type, in network byte order, followed by one or more bytes representing the actual identifier:

```
< 16 bits >     DHCP identifier used
< n bytes >     MD5 digest
```

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 RFC2535[7]. 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 Type Codes

The type code can have one of three classes of values. The first class contains just the value zero. This type indicates that the remaining contents of the DHCID record encode an identifier that is based on the client's link-layer network address.

The second class of types contains just the value 0xFFFF. This type code is reserved for future extensibility.

The third class of types contains all the values not included in the first two - that is, every value other than zero or 0xFFFF. Types in this class indicate that the remaining contents of the DHCID record encode an identifier that is based on the DHCP option whose code is the same as the specified type. The most common value in this class at the time of the writing of this specification is 0x3d (61 decimal), which is the DHCP option code for the Client Identifier option [8].

3.4 Computation of the RDATA

The DHCID RDATA is formed by concatenating the two type bytes with some variable-length identifying data.

< type > < data >

The RDATA for all type codes other than 0xffff, which is reserved for future expansion, is formed by concatenating the two type bytes and a 16-byte MD5 hash value. The input to the hash function is defined to be:

data = MD5(< identifier > < FQDN >)

The FQDN is represented in the buffer in unambiguous canonical form as described in RFC2535[7], section 8.1. The type code and the identifier are related as specified in Section 3.3: the type code describes the source of the identifier.

type code	identifier
0x0000	htype,hlen,chaddr from the client's DHCPREQUEST
0x0001-0xfffe	'data' portion of a DHCP option from the client's DHCPREQUEST
0xffff	RESERVED

The "Resolution of DNS Name Conflicts"[1] specification describes the selection process that updaters follow to choose an identifier from the information presented in a client's DHCPREQUEST message.

When the updater is using the client's link-layer address as the identifier, the first two bytes of the DHCID RDATA MUST be zero. To generate the rest of the resource record, the updater computes a one-way hash using the MD5 algorithm across a buffer containing the client's network hardware type, link-layer address, and the FQDN data. Specifically, the first byte 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 bytes of the chaddr field in the client's DHCPREQUEST message follow, in the same order in which the bytes appear in the DHCPREQUEST message. The number of significant bytes in the 'chaddr' field is specified in the 'hlen' field of the DHCPREQUEST message. The FQDN data, as specified above, follows.

When the updater is using a DHCP option sent by the client in its DHCPREQUEST message, the first two bytes of the DHCID RR MUST be the option code of that option, in network byte order. For example, if the DHCP client identifier option is being used, the first byte of the DHCID RR should be zero, and the second byte should be 61 decimal. The rest of the DHCID RR MUST contain the results of computing an MD5 hash across the payload of the option being used, followed by the FQDN. The payload of a DHCP option consists of the

bytes of the option following the option code and length.

3.5 Examples

3.5.1 Example 1

A DHCP server allocating the IPv4 address 10.0.0.1 to a client with 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 bytes to zero, and performing an MD5 hash computation across a buffer containing the Ethernet MAC type byte, 0x01, the six bytes of MAC address, and the domain name (represented as specified in Section 3.4).

```
client.example.com.        A       10.0.0.1
client.example.com.        DHCID   AAAUMru0ZM5OK/PdVAJgZ/HU
```

3.5.2 Example 2

A DHCP server allocates the IPv4 address 10.0.12.99 to a client which 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 bytes to the option code, 0x003d, and performing an MD5 hash computation across a buffer containing the seven bytes from the client-id option and the FQDN (represented as specified in Section 3.4).

```
chi.example.com.   A       10.0.12.99
chi.example.com.   DHCID   AD3dquu0xNqYn/4zw2FXy8X3
```

4. Use of the DHCID RR

This RR MUST NOT be used for any purpose other than that detailed in "Resolution of DNS Name Conflicts"[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 avoid exposing private information about DHCP clients to public scrutiny, a one-way hash is used to obscure all client information. 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.

Administrators should be wary of permitting unsecured DNS updates to zones which are exposed to the global Internet. Both DHCP clients and servers SHOULD use some form of update authentication (e.g., TSIG[9]) when performing DNS updates.

7. IANA Considerations

IANA is requested to allocate an RR type number for the DHCID record type.

References

[1] Stapp, M., "Resolution of DNS Name Conflicts Among DHCP Clients (draft-ietf-dhc-dns-resolution-*)", March 2001.

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

[3] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, Mar 1997.

[4] Mockapetris, P., "Domain names - Concepts and Facilities", RFC 1034, Nov 1987.

[5] Mockapetris, P., "Domain names - Implementation and Specification", RFC 1035, Nov 1987.

[6] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, April 1992.

[7] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

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

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

Authors' Addresses

Mark Stapp
Cisco Systems, Inc.
250 Apollo Dr.
Chelmsford, MA 01824
USA

Phone: 978.244.8498
EMail: mjs@cisco.com

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

EMail: mellon@nominum.com

Andreas Gustafsson
Nominum, Inc.
950 Charter St.
Redwood City, CA 94063
USA

EMail: gson@nominum.com

Full Copyright Statement

Acknowledgement

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