DNSEXT Working Group M. Stapp
Internet-Draft Cisco Systems, Inc.
Expires: August 31, 2001 T. Lemon
A. Gustafsson
Nominum, Inc.
March 2, 2001

A DNS RR for Encoding DHCP Information
<draft-ietf-dnsext-dhcid-rr-02.txt>

Status of this Memo

Copyright Notice

Abstract

A situation can arise where multiple DHCP clients request the same DNS name from their (possibly distinct) DHCP servers. To resolve such conflicts, 'Resolution of DNS Name Conflicts'[6] proposes storing client identifiers in the DNS to unambiguously associate domain names with the DHCP clients "owning" them. This memo defines a distinct RR type for use by DHCP 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[1].

2. Introduction

A set of procedures to allow DHCP[2] clients and servers to automatically update the DNS (RFC1034[4], RFC1035[5]) is proposed in "Resolution of DNS Name Conflicts"[6].

A situation can arise where multiple DHCP clients wish to use the same DNS name. To resolve such conflicts, Resolution of DNS Name Conflicts[6] proposes storing client identifiers in the DNS to unambiguously associate domain names with the DHCP clients using them. In the interest of clarity, it would be preferable for this DHCP information to use a distinct RR type.

This memo defines a distinct RR type for this purpose for use by DHCP clients or servers, the "DHCID" RR.

3. The DHCID RR

The DHCID RR is defined with mnemonic DHCID and type code [TBD].

4. 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. In DNS master files, the RDATA is represented in base 64 encoding (see Appendix A (Section 7)) and may be divided up into any number of white space separated substrings, down to single base 64 digits, which are concatenated to obtain the full signature. These substrings can span lines using the standard parenthesis. This format is identical to that used for representing binary data in DNSSEC (RFC2535[7]).

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 safely perform dynamic DNS updates to the same zone. From the updater's perspective, the DHCID resource record consists of a 16-bit identifier type, followed by one or more bytes representing the actual identifier.

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 draft is 61, which is the DHCP option code[3] for the Client Identifier option.

The data following the type code (for type codes other than 0xFFFF) is derived by running a one-way hash across the identifying information. The details of this are specified in "Resolution of DNS Name Conflicts"[6].

This RR MUST NOT be used for any purpose other than that detailed in "Resolution of DNS Name Conflicts"[6]. Althought 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 [6].

4.1 Example

A DHCP server allocating the IPv4 address 10.0.0.1 to a client "client.org.nil" might use the client's link-layer address to identify the client:

```
client.org.nil.    A       10.0.0.1
client.org.nil.    DHCID   AAAYKREXIgqtwYgQo93/yNlJ
```

A DHCP server allocating the IPv4 address 10.0.12.99 to a client "chi.org.nil" might use the DHCP client identifier option to identify the client:

```
chi.org.nil.       A       10.0.12.99
chi.org.nil.       DHCID   AGGScSLaAYjdOhGMHKD/lJ2B
```

5. 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.

6. IANA Considerations

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

7. Appendix A: Base 64 Encoding

The following encoding technique is taken from RFC 2045[8] by N. Borenstein and N. Freed. It is reproduced here in an edited form for convenience.

A 65-character subset of US-ASCII is used, enabling 6 bits to be represented per printable character. (The extra 65th character, "=", is used to signify a special processing function.)

The encoding process represents 24-bit groups of input bits as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8-bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the base 64 alphabet.

Each 6-bit group is used as an index into an array of 64 printable characters. The character referenced by the index is placed in the output string.

The Base 64 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

Special processing is performed if fewer than 24 bits are available

at the end of the data being encoded. A full encoding quantum is always completed at the end of a quantity. When fewer than 24 input bits are available in an input group, zero bits are added (on the right) to form an integral number of 6-bit groups. Padding at the end of the data is performed using the '=' character. Since all base 64 input is an integral number of octets, only the following cases can arise: (1) the final quantum of encoding input is an integral multiple of 24 bits; here, the final unit of encoded output will be an integral multiple of 4 characters with no "=" padding, (2) the final quantum of encoding input is exactly 8 bits; here, the final unit of encoded output will be two characters followed by two "=" padding characters, or (3) the final quantum of encoding input is exactly 16 bits; here, the final unit of encoded output will be three characters followed by one "=" padding character.

References

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

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

[3] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, 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] Stapp, M., "Resolution of DNS Name Conflicts Among DHCP Clients (draft-ietf-dhc-dns-resolution-*)", July 2000.

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

[8] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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.