Network Working Group
Request for Comments: 3396
Updates: 2131
Category: Standards Track

T. Lemon
Nominum, Inc.
S. Cheshire
Apple Computer, Inc.

November 2002

Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)

Status of this Memo

Copyright Notice

Abstract

This document specifies the processing rules for Dynamic Host Configuration Protocol (DHCPv4) options that appear multiple times in the same message. Multiple instances of the same option are generated when an option exceeds 255 octets in size (the maximum size of a single option) or when an option needs to be split apart in order to take advantage of DHCP option overloading. When multiple instances of the same option appear in the options, file and/or sname fields in a DHCP packet, the contents of these options are concatenated together to form a single option prior to processing.

1. Introduction

This document updates RFC 2131 [3] by clarifying the rules for option concatenation specified in section 4.1. It is expected that the reader will be familiar with this portion of RFC 2131. The text in section 4.1 that reads "Options may appear only once, unless otherwise specified in the options document." should be considered as deleted.

The DHCP protocol [3] specifies objects called "options" that are encoded in the DHCPv4 packet to pass information between DHCP protocol agents. These options are encoded as a one-byte type code, a one-byte length, and a buffer consisting of the number of bytes specified in the length, from zero to 255.

However, in some cases it may be useful to send options that are longer than 255 bytes. RFC 2131 [3] specifies that when more than one option with a given type code appears in the DHCP packet, all such options should be concatenated together. It does not, however, specify the order in which this concatenation should occur.

We specify here the ordering that MUST be used by DHCP protocol agents when sending options with more than 255 bytes. This method also MUST be used for splitting options that are shorter than 255 bytes, if for some reason the encoding agent needs to do so. DHCP protocol agents MUST use this method whenever they receive a DHCP packet containing more than one occurrence of a certain type of option.

2. Terminology

DHCP
: Throughout this document, the acronym "DHCP" is used to refer to the Dynamic Host Configuration Protocol as specified in RFC 2131 [3] and RFC 2132 [4].

DHCPv4
: We have used the term "DHCPv4" in the abstract for this document to distinguish between the DHCP protocol for IPv4 as defined in RFC 2131 and RFC 2132 and the DHCP protocol for IPv6, which, at the time that this document was written, was still under development.

DHCP protocol agents
: This refers to any device on the network that sends or receives DHCP packets - any DHCP client, server or relay agent. The nature of these devices is not important to this specification.

Encoding agent
: The DHCP protocol agent that is composing a DHCP packet to send.

Decoding agent
: The DHCP protocol agent that is processing a DHCP packet it has received.

Options
: DHCP options are collections of data with type codes that indicate how the options should be used. Options can specify information that is required for the DHCP protocol, IP stack configuration parameters for the client, information allowing the client to rendezvous with DHCP servers, and so on.

Option overload
The DHCP packet format is based on the BOOTP packet format defined in RFC 951 [1]. When used by DHCP protocol agents, BOOTP packets have three fields that can contain options. These are the optional parameters field, the sname field, and the filename field. The DHCP options specification [4] defines the DHCP Overload option, which specifies which of these three fields is actually being used in any given DHCP message to store DHCP options.

3. Requirements Language

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in BCP 14, RFC 2119 [2].

4. Applicability

This specification applies when a DHCP agent is encoding a packet containing options, where some of those options must be broken into parts. This need can occur for two reasons. First, it can occur because the value of an option that needs to be sent is longer than 255 bytes. In this case, the encoding agent MUST follow the algorithm specified here. It can also occur because there is not sufficient space in the current output buffer to store the option, but there is space for part of the option, and there is space in another output buffer for the rest. In this case, the encoding agent MUST either use this algorithm or not send the option at all.

This specification also applies in any case where a DHCP protocol agent has received a DHCP packet that contains more than one instance of an option of a given type. In this case, the agent MUST concatenate these separate instances of the same option in the way that we specify here.

This option updates the Dynamic Host Configuration Protocol [3] and DHCP Options and BOOTP vendor extensions [4] documents. However, because many currently-deployed DHCP protocol agents do not implement option concatenation, DHCP protocol agents should be careful not to transmit split options unless either it will not matter if the recipient cannot correctly reassemble the options, or it is certain that the recipient implements concatenation.

Let us divide all DHCP options into two categories - those that, by definition, require implementation of the mechanisms defined in this document, and those that do not. We will refer to the former as concatenation-requiring options, and the latter as non-concatenation-requiring options. In order for an option to be a

concatenation-requiring option, the protocol specification that defines that option must require implementation of option splitting and option concatenation as described in this document, by specifically referencing this document.

A DHCP protocol agent SHOULD NOT split an option as described in this document unless it has no choice, or it knows that its peer can properly handle split options. A peer is assumed to properly handle split options if it has provided or requested at least one concatenation-requiring option. Alternatively, the administrator of the agent generating the option can specifically configure the agent to assume that the recipient can correctly concatenate options split as described in this document.

Some implementors may find it easiest to only split concatenation-requiring options, and never split non-concatenation-requiring options. This is permissible. However, an implementation which supports any concatenation-requiring option MUST be capable of concatenating received options for both concatenation-requiring and non-concatenation-requiring options.

No restrictions apply to option concatenation when a DHCP agent receives a DHCP message. Any DHCP protocol agent that implements the mechanisms described in this document can assume that when it receives two options of the same type, it should concatenate them.

5. The Aggregate Option Buffer

DHCP options can be stored in the DHCP packet in three separate portions of the packet. These are the optional parameters field, the sname field, and the file field, as described in RFC 2131 [3]. This complicates the description of the option splitting mechanism because there are three separate fields into which split options may be placed.

To further complicate matters, an option that doesn’t fit into one field can’t overlap the boundary into another field - the encoding agent must instead break the option into two parts and store one part in each buffer.

To simplify this discussion, we will talk about an aggregate option buffer, which will be the aggregate of the three buffers. This is a logical aggregation - the buffers MUST appear in the locations in the DHCP packet described in RFC 2131 [3].

The aggregate option buffer is made up of the optional parameters field, the file field, and the sname field, in that order.

WARNING: This is not the physical ordering of these fields in the DHCP packet.

Options MUST NOT be stored in the aggregate option buffer in such a way that they cross either boundary between the three fields in the aggregate buffer.

The encoding agent is free to choose to use either or both the sname field and file field. If the encoding agent does not choose to use either or both of these two fields, then they MUST NOT be considered part of the aggregate option buffer in that case.

6. Encoding Agent Behavior

Encoding agents decide to split options based on the reasons we have described in the preceding section entitled "applicability".

Options can be split on any octet boundary. No split portion of an option that has been split can contain more than 255 octets. The split portions of the option MUST be stored in the aggregate option buffer in sequential order - the first split portion MUST be stored first in the aggregate option buffer, then the second portion, and so on. The encoding agent MUST NOT attempt to specify any semantic information based on how the option is split.

Note that because the aggregate option buffer does not represent the physical ordering of the DHCP packet, if an option were split into three parts and each part went into one of the possible option fields, the first part would go into the optional parameters field, the second part would go into the file field, and the third part would go into the sname field. This maintains consistency with section 4.1 of RFC 2131 [3].

Each split portion of an option MUST be stored in the aggregate option buffer as if it were a normal variable-length option as described in RFC 2132 [4]. The length fields of each split portion of the option MUST add up to the total length of the option data. For any given option being split, the option code field in each split portion MUST be the same.

7. Decoding Agent Behavior

When a decoding agent is scanning an incoming DHCP packet’s option buffer and finds two or more options with the same option code, it MUST consider them to be split portions of an option as described in the preceding section.

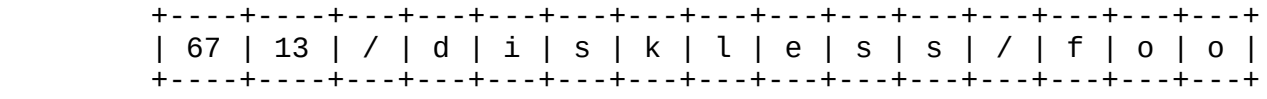
In the case that a decoding agent finds a split option, it MUST treat the contents of that option as a single option, and the contents MUST be reassembled in the order that was described above under encoding agent behavior.

The decoding agent should ensure that when the option's value is used, any alignment issues that are particular to the machine architecture on which the decoding agent is running are accounted for - there is no requirement that the encoding agent align the options in any particular way.

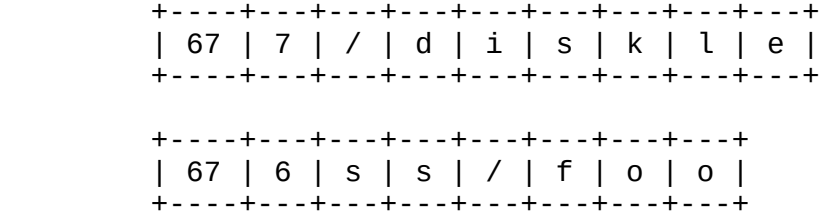
There is no semantic meaning to where an option is split - the encoding agent is free to split the option at any point, and the decoding agent MUST reassemble the split option parts into a single object, and MUST NOT treat each split portion of the option as a separate object.

8. Example

Consider an option, Bootfile name (option code 67), with a value of "/diskless/foo". Normally, this would be encoded as a single option, as follows:



If an encoding agent needed to split the option in order to fit it into the option buffer, it could encode it as two separate options, as follows, and store it in the aggregate option buffer in the following sequence:



9. Security Considerations

This document raises no new security issues. Potential exposures to attack in the DHCP protocol are discussed in section 7 of the DHCP protocol specification [3] and in Authentication for DHCP Messages [5].

Note that the authentication option itself can be split; in such cases implementations must be careful when setting the authentication field to zero (prior to generation or verification of the MAC) as it may be split across multiple options.

10. References

10.1. Normative References

[1] Croft, W. and J. Gilmore, "BOOTSTRAP PROTOCOL (BOOTP)", RFC 951, September 1985.

[2] Bradner, S., "Key words for use in RFCs to indicate requirement levels", BCP 14, RFC 2119, March 1997.

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

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

10.2. Informative References

[5] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

11. Intellectual Property Statement

12. Authors' Addresses

Ted Lemon
Nominum, Inc.
2385 Bay Road
Redwood City, CA 94043
USA

EMail: mellon@nominum.com

Stuart Cheshire
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 3207
EMail: rfc@stuartcheshire.org

13. Full Copyright Statement

Acknowledgement

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