fuzz_dhcpc6 Usage

Note: This document does not describe or explain DHCPv6, its messages or options. Consult RFC 3315 if you require that information.

Acknowledgement: Some code shamelessly stolen from Brandon Hutcheson and Graeme Neilson.

Network Interface

The network interface to use for receiving/sending DHCPv6 messages **must be** specified as the last (or only) command line argument. Typically this will be *eth0* but any active network interface should be useable.

Victim Selection Modes

A DHCPv6 client is identified by a combination of three types of data:

- IPv6 address
- Transaction ID – Used to synchronize responses
- Client ID – Optional DHCP Unique Identifier

All or none of these fields may be specified to select a victim. One of the following victim selection modes can be used:

fuzz_dhcpc6 eth0

Listen on ff02::1:2 for a client sending a DHCPv6 SOLICIT message. Extract client IPv6 address, Transaction ID and Client ID from the SOLICIT message and send fuzzed DHCPv6 messages to the client. The default message type is ADVERTISE and all message options specified in RFC 3315 are included one time in the order that they are described in RFC 3315. Currently the fuzz_dhcpc6 can only respond when Type 1 DUID is used for the Client ID.

fuzz_dhcpc6 –v fe80:21e:c9ff:fe61:6cca eth0

Send DHCPv6 ADVERTISE messages to the victim at IPv6 address fe80:21e:c9ff:fe61:6cca and fuzz Transaction ID and Client ID along with all DHCPv6 message options.

fuzz_dhcpc6 –v fe80:21e:c9ff:fe61:6cca –x 0x280e77 eth0

Send DHCPv6 ADVERTISE messages to victim at fe80:21e:c9ff:fe61:6cca with Transaction ID set to 0x280e77 and fuzz Client ID along with all DHCPv6 message options.

fuzz_dhcpc6 –v fe80:21e:c9ff:fe61:6cca –x 0x280e77 –c 000100011a62b677001ec9616cca eth0

Send DHCPv6 ADVERTISE messages to victim at fe80:21e:c9ff:fe61:6cca with Transaction ID set to 0x280e77 and Client ID set to 000100011a62b677001ec9616cca along with all fuzzed DHCPv6 message options.

Online Usage Help

fuzz_dhcpc6 -h

DHCPv6 Message Type Fuzzing

The DHCPv6 message type can be selected or fuzzed in any of the victim selection modes by including one of the following arguments on the command line:

Option	DCHPv6 Message Type
-1	SOLICIT
-2	ADVERTISE (default)
-3	REQUEST
-4	CONFIRM
-5	RENEW
-6	REBIND
-7	REPLY
-8	RELEASE
-9	DECLINE
-A	RECONFIGURE
-B	INFORMATION REQUEST
-C	RELAY-FORWARD
-D	RELAY-REPLY
-m	Fuzz the message type (0-255)

Message Options

By default, all DHCPv6 message options defined in RFC 3315 are included one time in the order described in the RFC. They are populated using data that is nonsense but allows them to be interpreted as valid RFC compliant options. However as the option contents are fuzzed, they will generate non-compliant options that will hopefully generate errors on the DHCP client.

Option Selection

DHCPv6 message options can be reordered, duplicated or omitted by defining the “-o” argument on the command line in any of the victim selection modes. Each DHCPv6 message option is assigned a unique lowercase letter as a tag that can be specified in the “–o” command line option to override the default message options. For example:

```
-oiaaq
```

Only includes OPTION_AUTH (i), followed by 2 copies of OPTION_SERVERID (a) then OPTION_RECONF_MSG (q). See ***Fuzzing Details*** for option tags, default values and fuzzing patterns used for each message option if fuzzed.

Note: The Message type, Transaction ID and Client ID are not selectable using the “-o” command line argument. Instead use “-m” to fuzz message type, “-x0” to fuzz Transaction ID and “-c0” to fuzz Client ID. Note that “-x” and “-c” arguments are only used with the “-v” argument.

Fuzzing Patterns

The fuzz_dhcpc6 program uses four fuzzing patterns for DHCPv6 message data based on the type of data being fuzzed. These are 2 variations of one byte data fields, one for word data fields (2 byte data) and one for double word data fields (4 byte).

Pattern	Size	Values
B	1	0, 1, 254, 255
X	1	0 – 255
W	2	0x0000, 0x0001, 0x00fe, 0x00ff, 0x0100, 0x0101, 0x01fe, 0x01ff, 0xfe00, 0xfe01, 0xfefe, 0xfeff, 0xff00, 0xff01, 0xfffe, 0xffff
D	4	0x00000000, 0x00000001, 0x000000fe, 0x000000ff, 0x00000100, 0x00000101, 0x000001fe, 0x000001ff, 0x0000fe00, 0x0000fe01, 0x0000fefe, 0x0000feff, 0x0000ff00, 0x0000ff01, 0x0000fffe, 0x0000ffff, 0x00010000, 0x00010001, 0x000100fe, 0x000100ff, 0x00010100, 0x00010101, 0x000101fe, 0x000101ff, 0x0001fe00, 0x0001fe01, 0x0001fefe, 0x0001feff, 0x0001ff00, 0x0001ff01, 0x0001fffe, 0x0001ffff, 0x00fe0000, 0x00fe0001, 0x00fe00fe, 0x00fe00ff, 0x00fe0100, 0x00fe0101, 0x00fe01fe, 0x00fe01ff, 0x00fefe00, 0x00fefe01, 0x00fefefe, 0x00fefeff, 0x00feff00, 0x00feff01, 0x00fefffe, 0x00feffff, 0x00ff0000, 0x00ff0001, 0x00ff00fe, 0x00ff00ff, 0x00ff0100, 0x00ff0101, 0x00ff01fe, 0x00ff01ff, 0x00fffe00, 0x00fffe01, 0x00fffefe, 0x00fffeff, 0x00ffff00, 0x00ffff01, 0x00fffffe, 0x00ffffff, 0x01000000, 0x01000001, 0x010000fe, 0x010000ff, 0x01000100, 0x01000101, 0x010001fe, 0x010001ff, 0x0100fe00, 0x0100fe01, 0x0100fefe, 0x0100feff, 0x0100ff00, 0x0100ff01, 0x0100fffe, 0x0100ffff, 0x01010000, 0x01010001, 0x010100fe, 0x010100ff, 0x01010100, 0x01010101, 0x010101fe, 0x010101ff, 0x0101fe00, 0x0101fe01, 0x0101fefe, 0x0101feff, 0x0101ff00, 0x0101ff01, 0x0101fffe, 0x0101ffff, 0x01fe0000, 0x01fe0001, 0x01fe00fe, 0x01fe00ff, 0x01fe0100, 0x01fe0101, 0x01fe01fe, 0x01fe01ff, 0x01fefe00, 0x01fefe01, 0x01fefefe, 0x01fefeff, 0x01feff00, 0x01feff01, 0x01fefffe, 0x01feffff, 0x01ff0000, 0x01ff0001, 0x01ff00fe, 0x01ff00ff, 0x01ff0100, 0x01ff0101, 0x01ff01fe, 0x01ff01ff, 0x01fffe00, 0x01fffe01, 0x01fffefe, 0x01fffeff, 0x01ffff00, 0x01ffff01, 0x01fffffe, 0x01ffffff, 0xfe000000, 0xfe000001, 0xfe0000fe, 0xfe0000ff, 0xfe000100, 0xfe000101, 0xfe0001fe, 0xfe0001ff, 0xfe00fe00, 0xfe00fe01, 0xfe00fefe, 0xfe00feff, 0xfe00ff00, 0xfe00ff01, 0xfe00fffe, 0xfe00ffff, 0xfe010000, 0xfe010001, 0xfe0100fe, 0xfe0100ff, 0xfe010100, 0xfe010101, 0xfe0101fe, 0xfe0101ff, 0xfe01fe00, 0xfe01fe01, 0xfe01fefe, 0xfe01feff, 0xfe01ff00, 0xfe01ff01, 0xfe01fffe, 0xfe01ffff, 0xfefe0000, 0xfefe0001, 0xfefe00fe, 0xfefe00ff, 0xfefe0100, 0xfefe0101, 0xfefe01fe, 0xfefe01ff, 0xfefefe00, 0xfefefe01, 0xfefefefe, 0xfefefeff, 0xfefeff00, 0xfefeff01, 0xfefefffe, 0xfefeffff, 0xfeff0000, 0xfeff0001, 0xfeff00fe, 0xfeff00ff, 0xfeff0100, 0xfeff0101, 0xfeff01fe, 0xfeff01ff, 0xfefffe00, 0xfefffe01, 0xfefffefe, 0xfefffeff, 0xfeffff00, 0xfeffff01, 0xfefffffe, 0xfeffffff, 0xff000000, 0xff000001, 0xff0000fe, 0xff0000ff, 0xff000100, 0xff000101, 0xff0001fe, 0xff0001ff, 0xff00fe00, 0xff00fe01, 0xff00fefe, 0xff00feff, 0xff00ff00, 0xff00ff01, 0xff00fffe, 0xff00ffff, 0xff010000, 0xff010001, 0xff0100fe, 0xff0100ff, 0xff010100, 0xff010101, 0xff0101fe, 0xff0101ff, 0xff01fe00, 0xff01fe01, 0xff01fefe, 0xff01feff, 0xff01ff00, 0xff01ff01, 0xff01fffe, 0xff01ffff, 0xfffe0000, 0xfffe0001, 0xfffe00fe, 0xfffe00ff, 0xfffe0100, 0xfffe0101, 0xfffe01fe, 0xfffe01ff, 0xfffefe00, 0xfffefe01, 0xfffefefe, 0xfffefeff, 0xfffeff00, 0xfffeff01, 0xfffefffe, 0xfffeffff, 0xffff0000, 0xffff0001, 0xffff00fe, 0xffff00ff, 0xffff0100, 0xffff0101, 0xffff01fe, 0xffff01ff, 0xfffffe00, 0xfffffe01, 0xfffffefe, 0xfffffeff, 0xffffff00, 0xffffff01, 0xfffffffe, 0xffffffff

For each DHCPv6 message sent, only one field of one option is fuzzed at a time. That is the first message sent will only fuzz the first field of the first option (or Message Type, Transaction ID, Client ID if any of them are being fuzzed) and the last message sent with fuzz the last field of the last option specified. If the default value of a field being fuzzed matches one of the fuzzing values, that fuzzing value is skipped since it will be sent as the default value when all other options are being fuzzed. In addition, after all fuzzing values have been sent, the original value of the field is XORed with all zeroes and all ones.

Fuzzing Details

DHCPv6 Field Or Option	-o Tag	Data Field – Default (Fuzzing Pattern)	Fuzz Count
DHCP Message Type	N/A	Message type – 0x2 (X)	256
Transaction ID	N/A	Transaction ID - 0x0 0x0 0x0 (BBB)	9
OPTION_CLIENTID	N/A	DUID type – 0x0 0x1 (W) Hardware Type - 0x0 0x1 (W) Time - 0x1A 0x62 0xB6 0x77 (D) Link-layer Address – MAC of eth0 (WWW)	330
OPTION_SERVERID	a	DUID type - 0x0 0x1 (W) Hardware Type - 0x0 0x1 (W) Time - 0x1A 0x62 0xB6 0x77 (D) Link-layer Address – MAC of eth0 (WWW)	342
OPTION_IA_NA	b	IAID – 0x1 0x2 0x3 0x4 (D) T1 – 0x0 0x0 0x0 0x0 (D) T2 – 0x0 0x0 0x0 0xA (D)	771
OPTION_IA_TA	c	IAID – 0x1 0x2 0x3 0x4 (D) Option – 0x0 0xD (W) Length – 0x0 0x4 (W) Status Code – 0x0 0x0 (W) Status Message – “OK” (W)	327
OPTION_IAADR	d	IPv6 Address – 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xFF 0xFF 0xFF 0xFF(D13-16) Preferred-lifetime – 0xFF 0xFF 0xFF 0xFF (D) Valid-lifetime – 0x0 0x0 0x0 0xD	766
OPTION_ORO	e	Requested Option Code 1 – 0x0 0x8 (XX)	512
OPTION_PREFERENCE	f	Preference Value – 0x0 (X)	256
OPTION_ELAPSED_TIME	g	Elapsed-time – 0x10 0xFF (BB)	9
OPTION_RELAY_MSG	h	msg-type – 0xC (B) hop-count – 0xF (B) link-address – “LINK ADDRESS” 0xD 0xE 0xF 0x10 (B16) peer-address - “PEER ADDRESS” 0xD 0xE 0xF 0x10	18
OPTION_AUTH	i	Protocol – 0x0 (B) Algorithm – 0x1 (B) RDM – 0x0 (B) Replay Detection – “REPLAY” 0x7 0x8 (DD) DHCP Realm – “REALM” Key ID – “KEY” 0x4 0x5 (DB) HMAC-MD5 – “HMAC-MD5” 0x9 0xA, 0xB, 0xC, 0xD, 0xE, 0xF, 0x10	789
OPTION_UNICAST	j	server-address – UNICAST ADDRESS 0x10 (DDDD)	1032
OPTION_STATUS_CODE	k	status-code – 0x0 0x0 (XX) status-message – “STATUS MESSAGE”	512
OPTION_RAPID_COMMIT	l	option-len – 0x0 (B)	3
OPTION_USER_CLASS	m	user-class-data – “USER” (D)	258
OPTION_VENDOR_CLASS	n	enterprise-number – 0x0 0x0 0x0 0x0 (D) vendor-class-len – 0x0 0x5 opaque-data* - “CLASS”	255
OPTION_VENDOR_OPTS	o	enterprise-number – 0x0 0x0 0x0 0x0 (D) opt-code – 0x0 0x0 option-len – 0x 0x6 option-data – “OPTION”	255
OPTION_INTERFACE_ID	p	interface-id – “INTERFACE ID” 0x20 0x20 (B14)	6
OPTION_RECONF_MSG	q	msg-type – 0x0 (X)	256
OPTION_RECONF_ACCEPT	r	Illegal field – 0x0 (B)	3