Distributed-Denial-of-Service Open Threat Signaling (DOTS) Server Discovery
draft-ietf-dots-server-discovery-03

Abstract

   It may not be possible for a network to determine the cause for an
   attack, but instead just realize that some resources seem to be under
   attack. To fill that gap, Distributed-Denial-of-Service Open Threat
   Signaling (DOTS) allows a network to inform a DOTS server that it is
   under a potential attack so that appropriate mitigation actions are
   undertaken.

   This document specifies mechanisms to configure DOTS clients with
   DOTS servers. The discovery procedure also covers the DOTS Signal
   Channel Call Home.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on December 2, 2019.

Copyright Notice

Table of Contents

1. Introduction

   DDoS Open Threat Signaling (DOTS) [I-D.ietf-dots-architecture]
   specifies an architecture, in which a DOTS client can inform a DOTS
   server that the network is under a potential attack and that
   appropriate mitigation actions are required. Indeed, because the
   lack of a common method to coordinate a real-time response among
   involved actors and network domains inhibits the effectiveness of
   DDoS attack mitigation, DOTS signal channel protocol
   [I-D.ietf-dots-signal-channel] is meant to carry requests for DDoS
   attack mitigation, thereby reducing the impact of an attack and
   leading to more efficient defensive actions in various deployment
   scenarios such as those discussed in [I-D.ietf-dots-use-cases].
   Moreover, DOTS clients can instruct a DOTS server to install
   filtering rules by means of DOTS data channel
   [I-D.ietf-dots-data-channel].

   The basic high-level DOTS architecture is illustrated in Figure 1
   ([I-D.ietf-dots-architecture]):

```
        +-----------+               +-------------+
        | Mitigator | ~~~~~~~~~~ | DOTS Server |
        +-----------+               +-------------+
                                          |
                                          |
                                          |
        +---------------+         +-------------+
        | Attack Target | ~~~~~ | DOTS Client |
        +---------------+         +-------------+
```
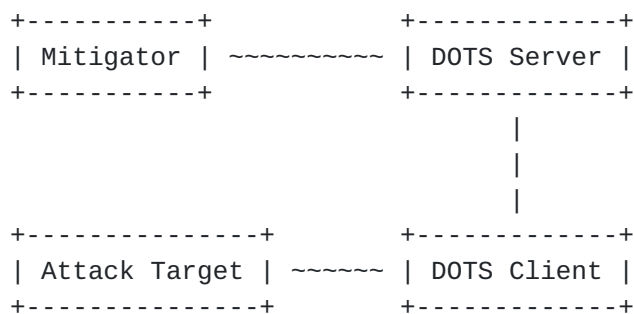
                   Figure 1: Basic DOTS Architecture

   [I-D.ietf-dots-architecture] specifies that the DOTS client may be
   provided with a list of DOTS servers; each associated with one or
   more IP addresses. These addresses may or may not be of the same
   address family. The DOTS client establishes one or more DOTS
   sessions by connecting to the provided DOTS server addresses.

   This document specifies methods for DOTS clients to discover their
   DOTS server(s). The rationale for specifying multiple discovery
   mechanisms is discussed in Section 3.

   The discovery methods can also used by a DOTS server to locate a DOTS
   client in the context of DOTS Signal Channel Call Home
   [I-D.ietf-dots-signal-call-home].

Considerations for the selection of DOTS server(s) by multi-homed
DOTS clients is out of scope; the reader should refer to
[I-D.ietf-dots-multihoming] for more details.

Likewise, happy eyeballs considerations for DOTS signal channel
protocol are out of scope. The reader should refer to Section 4 of
[I-D.ietf-dots-signal-channel].

This document assumes that security credentials to authenticate DOTS
server(s) are provisioned to a DOTS client using a variety of means
such as (but not limited to) those discussed in
[I-D.ietf-netconf-zerotouch] or
[I-D.ietf-anima-bootstrapping-keyinfra]. DOTS clients use those
credentials for authentication purposes following the rules
documented in [I-D.ietf-dots-signal-channel].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP
14 [RFC2119][RFC8174] when, and only when, they appear in all
capitals, as shown here.

The reader should be familiar with the terms defined in
[I-D.ietf-dots-architecture] and [RFC3958].

DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC8415].

"Peer DOTS agent" refers to the peer DOTS server (normal DOTS
operation) or to a peer DOTS client (for DOTS Signal Channel Call
Home).

3. Why Multiple Discovery Mechanisms?

It is tempting to specify one single discovery mechanism for DOTS.
Nevertheless, the analysis of the various use cases sketched in
[I-D.ietf-dots-use-cases] reveals that it is unlikely that one single
discovery method can be suitable for all the sample deployments.
Concretely:

o Many use cases discussed in [I-D.ietf-dots-use-cases] do involve a
   CPE device. Multiple CPEs, connected to distinct network
   providers may even be considered. It is intuitive to leverage on
   existing mechanisms such as discovery using service resolution or
   DHCP to provision the CPE acting as a DOTS client with the DOTS
   server(s).

o Resolving a DOTS server domain name offered by an upstream transit
   provider provisioned to a DOTS client into IP address(es) require
   the use of the appropriate DNS resolvers; otherwise, resolving
   those names will fail. The use of protocols such as DHCP does
   allow to associate provisioned DOTS server domain names with a
   list of DNS servers to be used for name resolution. Furthermore,
   DHCP allows to directly provision IP addresses avoiding therefore
   the need for extra lookup delays.

o Some of the use cases may allow DOTS clients to have direct
   communications with upstream DOTS servers; that is no DOTS gateway
   is involved. Leveraging on existing features that do not require
   specific feature on the node embedding the DOTS client may ease
   DOTS deployment. Typically, the use of Straightforward-Naming
   Authority Pointer (S-NAPTR) lookups [RFC3958] allows the DOTS
   server administrators to provision the preferred DOTS transport
   protocol between the DOTS client and the DOTS server and allows
   the DOTS client to discover this preference.

o The upstream network provider is not the DDoS mitigation provider
   for some of these use cases. It is safe to assume that for such
   deployments, the DOTS server(s) domain name is provided during the
   service subscription (i.e., manual/local configuration).

o Multiple DOTS clients may be enabled within a network (e.g.,
   enterprise network). Dynamic means to discover DOTS servers in a
   deterministic manner are interesting from an operational
   standpoint.

o Some of the use cases may involve a DOTS gateway that is
   responsible for selecting the appropriate DOTS server(s) to relay
   requests received from DOTS clients.

Consequently, this document describes a unified discovery logic
(Section 4) which involves the following mechanisms:

o Dynamic discovery using DHCP (Section 5).

o A resolution mechanism based on straightforward Naming Authority
   Pointer (S-NAPTR) resource records in the Domain Name System (DNS)
   (Section 6).

o DNS Service Discovery (Section 7).

4. **Unified DOTS Discovery Procedure**

   A key point in the deployment of DOTS is the ability of network
   operators to be able to configure DOTS clients with the correct DOTS
   server(s) information consistently. To accomplish this, operators
   will need a consistent set of ways in which DOTS clients can discover
   this information, and a consistent priority among these options. If
   some devices prefer manual configuration over dynamic discovery,
   while others prefer dynamic discovery over manual configuration, the
   result will be a process of "whack-a-mole", where the operator must
   find devices that are using the wrong DOTS server(s), determine how
   to ensure the devices are configured properly, and then reconfigure
   the device through the preferred method.

   All DOTS clients MUST support at least one of the three mechanisms
   below to determine a DOTS server list. All DOTS clients SHOULD
   implement all three, or as many as are practical for any specific
   device, of these ways to discover DOTS servers, in order to
   facilitate the deployment of DOTS in large scale environments:

   1. Explicit configuration:

       * Local/Manual configuration: A DOTS client, will learn the DOTS
          server(s) by means of local or manual DOTS configuration
          (i.e., DOTS servers configured at the system level).
          Configuration discovered from a DOTS client application is
          considered as local configuration.

          An implementation may give the user an opportunity (e.g., by
          means of configuration file options or menu items) to specify
          DOTS server(s) for each address family. These MAY be
          specified either as IP addresses or the DNS name of a DOTS
          server. When only DOTS server's IP addresses are configured,
          a reference identifier must also be configured for
          authentication purposes.

       * Automatic configuration (e.g., DHCP, an automation system):
          The DOTS client attempts to discover DOTS server(s) names and/
          or addresses from DHCP, as described in Section 5.

   2. Service Resolution : The DOTS client attempts to discover DOTS
       server name(s) using service resolution, as specified in
       Section 6.

   3. DNS SD: DNS Service Discovery. The DOTS client attempts to
       discover DOTS server name(s) using DNS service discovery, as
       specified in Section 7.

Some of these mechanisms imply the use of DNS to resolve the IP
address(es) of the DOTS server, while others imply an IP address of
the relevant DOTS server is obtained directly. Implementation
options may vary on a per device basis, as some devices may not have
DNS capabilities and/or proper configuration.

DOTS clients will prefer information received from the discovery
methods in the order listed.

On hosts with more than one interface or address family (IPv4/v6),
the DOTS server discovery procedure has to be performed for each
combination of interface and address family. A client MAY choose to
perform the discovery procedure only for a desired interface/address
combination if the client does not wish to discover a DOTS server for
all combinations of interface and address family.

The above procedure MUST also be followed by a DOTS gateway.
Likewise, this procedure MUST be followed by a DOTS server in the
context of DOTS Signal Channel Call Home
[I-D.ietf-dots-signal-call-home].

The discovery method MUST be reiterated upon the following events:

o Expiry of a lease associated with a discovered DOTS server.

o Expiry of a DOTS server's certificate currently in use.

o Attachment to a new network.

5. DHCP Options for DOTS

As reported in Section 1.7.2 of [RFC6125]:

   "few certification authorities issue server certificates based on
   IP addresses, but preliminary evidence indicates that such
   certificates are a very small percentage (less than 1%) of issued
   certificates".

In order to allow for PKIX-based authentication between a DOTS client
and server while accommodating for the current best practices for
issuing certificates, this document allows for configuring names to
DOTS clients. These names can be used for two purposes: to retrieve
the list of IP addresses of a DOTS server or to be presented as a
reference identifier for authentication purposes.

Defining the option to include a list of IP addresses would avoid a
dependency on an underlying name resolution, but that design requires
to also supply a name for PKIX-based authentication purposes.

The design assumes that the same peer DOTS agent is used for
establishing both signal and data channels. For more customized
configurations (e.g., transport-specific configuration, distinct DOTS
servers for the signal and the data channels), an operator can supply
only a DOTS reference identifier that will be then passed to the
procedure described in Section 6.

5.1. DHCPv6 DOTS Options

5.1.1. Format of DOTS Reference Identifier Option

The DHCPv6 DOTS option is used to configure a name of the DOTS server
(or the name of the DOTS client for DOTS Signal Channel Call Home).
The format of this option is shown in Figure 2.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_V6_DOTS_RI          |          Option-length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     dots-agent-name (FQDN)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: DHCPv6 DOTS Reference Identifier Option

The fields of the option shown in Figure 2 are as follows:

o Option-code: OPTION_V6_DOTS_RI (TBA1, see Section 9.1)
o Option-length: Length of the dots-server-name field in octets.
o dots-agent-name: A fully qualified domain name of the peer DOTS
   agent. This field is formatted as specified in Section 10 of
   [RFC8415].

An example of the dots-agent-name encoding is shown in Figure 3.
This example conveys the FQDN "dots.example.com.".

```
+------+------+------+------+------+------+------+------+------+
| 0x04 |  d   |  o   |  t   |  s   | 0x07 |  e   |  x   |  a   |
+------+------+------+------+------+------+------+------+------+
|  m   |  p   |  l   |  e   | 0x03 |  c   |  o   |  m   | 0x00 |
+------+------+------+------+------+------+------+------+------+
```
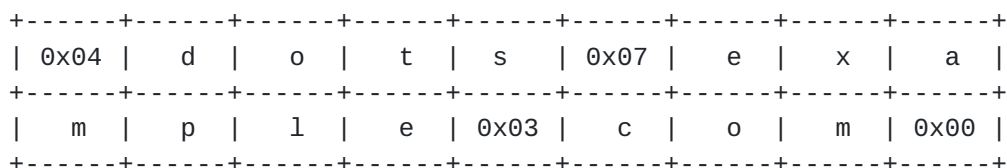
Figure 3: An example of the dots-agent-name Encoding

5.1.2. Format of DOTS Address Option

The DHCPv6 DOTS option can be used to configure a list of IPv6
addresses of a DOTS server (or a DOTS client for DOTS Signal Channel
Call Home). The format of this option is shown in Figure 4. As a
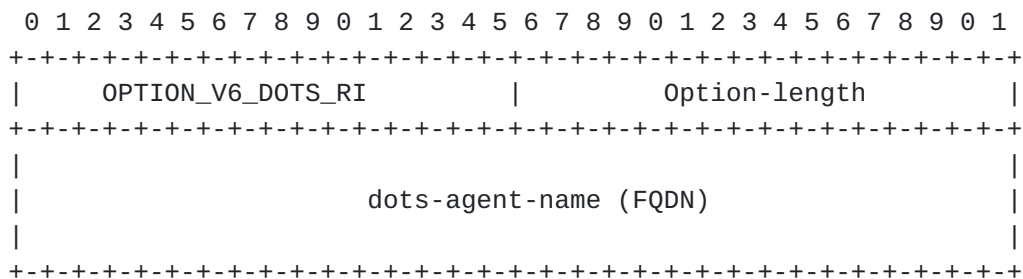reminder, this format follows the guidelines for creating new DHCPv6
options (Section 5.1 of [RFC7227]).

```
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   OPTION_V6_DOTS_ADDRESS        |         Option-length        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                      DOTS ipv6-address                        |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                      DOTS ipv6-address                        |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             ...                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
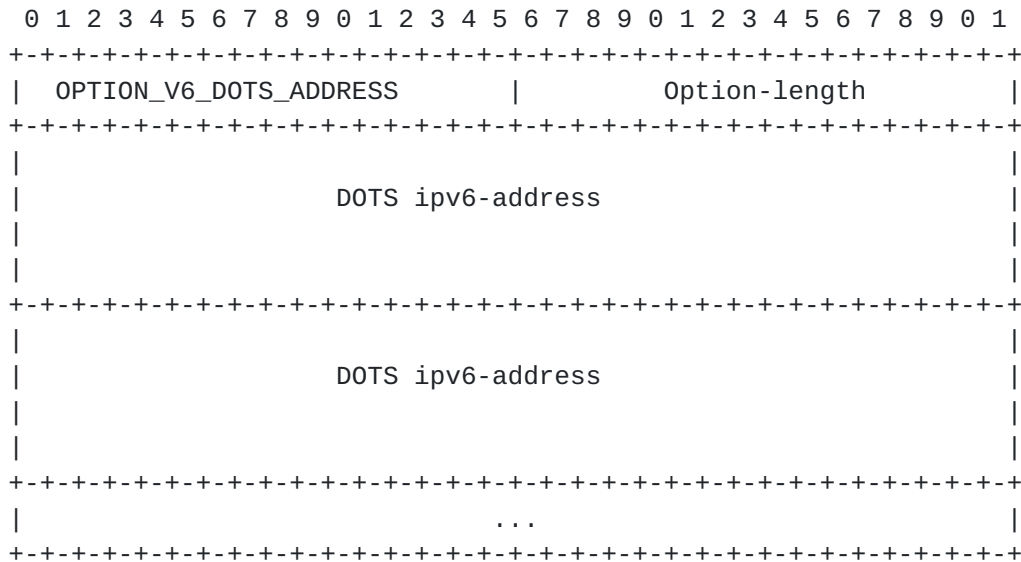
Figure 4: DHCPv6 DOTS Address Option

The fields of the option shown in Figure 4 are as follows:

o Option-code: OPTION_V6_DOTS_ADDRESS (TBA2, see Section 9.1)
o Option-length: Length of the 'DOTS ipv6-address(es)' field in
   octets. MUST be a multiple of 16.
o DOTS ipv6-address: Includes one or more IPv6 addresses [RFC4291]
   of the peer DOTS agent to be used by a DOTS agent for establishing
   a DOTS session.

   Note, IPv4-mapped IPv6 addresses (Section 2.5.5.2 of [RFC4291])
   are allowed to be included in this option.

To return more than one DOTS agents to the requesting DHCPv6 client,
the DHCPv6 server returns multiple instances of OPTION_V6_DOTS.

5.1.3. DHCPv6 Client Behavior

DHCP clients MAY request options OPTION_V6_DOTS_RI and
OPTION_V6_DOTS_ADDRESS, as defined in [RFC8415], Sections 18.2.1,
18.2.2, 18.2.4, 18.2.5, 18.2.6, and 21.7. As a convenience to the

   reader, it is mentioned here that the DHCP client includes the
   requested option codes in the Option Request Option.

   If the DHCP client receives more than one instance of
   OPTION_V6_DOTS_RI (or OPTION_V6_DOTS_ADDRESS) option, it MUST use
   only the first instance of that option.

   If the DHCP client receives both OPTION_V6_DOTS_RI and
   OPTION_V6_DOTS_ADDRESS, the content of OPTION_V6_DOTS_RI is used as
   reference identifier for authentication purposes (e.g., PKIX
   [RFC6125]), while the addresses included in OPTION_V6_DOTS_ADDRESS
   are used to reach the peer DOTS agent. In other words, the name
   conveyed in OPTION_V6_DOTS_RI MUST NOT be passed to underlying
   resolution library in the presence of OPTION_V6_DOTS_ADDRESS in a
   response.

   If the DHCP client receives OPTION_V6_DOTS_RI only, but
   OPTION_V6_DOTS_RI option contains more than one name, as
   distinguished by the presence of multiple root labels, the DHCP
   client MUST use only the first name. Once the name is validated
   (Section 8 of [RFC8415]), the name is passed to a name resolution
   library. Moreover, that name is also used as a reference identifier
   for authentication purposes.

   If the DHCP client receives OPTION_V6_DOTS_ADDRESS only, the
   address(es) included in OPTION_V6_DOTS_ADDRESS is used to reach the
   peer DOTS agent. In addition, these addresses can be used as
   identifiers for authentication.

   The DHCP client MUST silently discard multicast and host loopback
   addresses [RFC6890] conveyed in OPTION_V6_DOTS_ADDRESS.

5.2. DHCPv4 DOTS Options

5.2.1. Format of DOTS Reference Identifier Option

   The DHCPv4 DOTS option is used to configure a name of the peer DOTS
   agent. The format of this option is illustrated in Figure 5.

```
      Code   Length   Peer DOTS agent name
     +-----+-----+-----+-----+-----+-----+-----+--
     |TBA3 |  n  |  s1 |  s2 |  s3 |  s4 |  s5 |  ...
     +-----+-----+-----+-----+-----+-----+-----+--
```

   The values s1, s2, s3, etc. represent the domain name labels in the
   domain name encoding.

              Figure 5: DHCPv4 DOTS Reference Identifier Option

   The fields of the option shown in Figure 5 are as follows:

   o Code: OPTION_V4_DOTS_RI (TBA3, see Section 9.2);
   o Length: Includes the length of the "DOTS server name" field in
      octets; the maximum length is 255 octets.
   o Peer DOTS agent name: The domain name of the peer DOTS agent.
      This field is formatted as specified in Section 10 of [RFC8415].

5.2.2. Format of DOTS Address Option

   The DHCPv4 DOTS option can be used to configure a list of IPv4
   addresses of a peer DOTS agent. The format of this option is
   illustrated in Figure 6.

```
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Code=TBA4    |     Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | List-Length   |    List of     |
    +-+-+-+-+-+-+-+-+     DOTS        |
    /         IPv4 Addresses         /
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ---
    | List-Length   |    List of     |    |
    +-+-+-+-+-+-+-+-+     DOTS        |    |
    /         IPv4 Addresses         /    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |
    .            ...                 . optional
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |
    | List-Length   |    List of     |    |
    +-+-+-+-+-+-+-+-+     DOTS        |    |
    /         IPv4 Addresses         /    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ---
```


                 Figure 6: DHCPv4 DOTS Address Option

   The fields of the option shown in Figure 6 are as follows:

o Code: OPTION_V4_DOTS_ADDRESS (TBA4, see Section 9.2);

o Length: Length of all included data in octets. The minimum length
   is 5.

o List-Length: Length of the "List of DOTS IPv4 Addresses" field in
   octets; MUST be a multiple of 4.

o List of DOTS IPv4 Addresses: Contains one or more IPv4 addresses
   of the peer DOTS agent to be used by a DOTS agent. The format of
   this field is shown in Figure 7.

o OPTION_V4_DOTS_ADDRESS can include multiple lists of DOTS IPv4
   addresses; each list is treated separately as it corresponds to a
   given peer DOTS agent.

   When several lists of DOTS IPv4 addresses are to be included,
   "List-Length" and "DOTS IPv4 Addresses" fields are repeated.

```
0     8     16    24    32    40    48
+-----+-----+-----+-----+-----+-----+--
|  a1 |  a2 |  a3 |  a4 |  a1 |  a2 | ...
+-----+-----+-----+-----+-----+-----+--
     IPv4 Address 1          IPv4 Address 2 ...
```

This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

           Figure 7: Format of the List of DOTS IPv4 Addresses

OPTION_V4_DOTS_ADDRESS is a concatenation-requiring option. As such,
the mechanism specified in [RFC3396] MUST be used if
OPTION_V4_DOTS_ADDRESS exceeds the maximum DHCPv4 option size of 255
octets.

5.2.3. DHCPv4 Client Behavior

To discover a peer DOTS agent, the DHCPv4 client MUST include both
OPTION_V4_DOTS_RI and OPTION_V4_DOTS_ADDRESS in a Parameter Request
List Option [RFC2132].

If the DHCP client receives more than one instance of
OPTION_V4_DOTS_RI (or OPTION_V4_DOTS_ADDRESS) option, it MUST use
only the first instance of that option.

If the DHCP client receives both OPTION_V4_DOTS_RI and
OPTION_V4_DOTS_ADDRESS, the content of OPTION_V6_DOTS_RI is used as
reference identifier for authentication purposes, while the addresses
included in OPTION_V4_DOTS_ADDRESS are used to reach the peer DOTS
agent. In other words, the name conveyed in OPTION_V4_DOTS_RI MUST
NOT be passed to underlying resolution library in the presence of
OPTION_V4_DOTS_ADDRESS in a response.

If the DHCP client receives OPTION_V4_DOTS_RI only, but
OPTION_V4_DOTS_RI option contains more than one name, as
distinguished by the presence of multiple root labels, the DHCP
client MUST use only the first name. Once the name is validated
(Section 10 of [RFC8415]), the name is passed to a name resolution
library. Moreover, that name is also used as a reference identifier
for authentication purposes.

If the DHCP client receives OPTION_V4_DOTS_ADDRESS only, the
address(es) included in OPTION_V4_DOTS_ADDRESS is used to reach the
peer DOTS server. In addition, these addresses can be used as
identifiers for authentication.

The DHCP client MUST silently discard multicast and host loopback
addresses conveyed in OPTION_V4_DOTS_ADDRESS.

6. Discovery using Service Resolution

This mechanism is performed in two steps:

1. A DNS domain name is retrieved for each combination of interface
    and address family. A DOTS client has to determine the domain in
    which it is located relying on dynamic means such as DHCP
    (Section 5) . Implementations MAY allow the user to specify a
    default name that is used, if no specific name has been
    configured.

2. Retrieved DNS domain names are then used for S-NAPTR lookups.
    Further DNS lookups may be necessary to determine DOTS server IP
    address(es).

Once the DOTS client has retrieved client's DNS domain or discovered
the peer DOTS agent name that needs to be resolved (e.g., Section 5),
an S-NAPTR lookup with 'DOTS' application service and the desired
protocol tag is made to obtain information necessary to connect to
the authoritative DOTS server within the given domain.

This specification defines "DOTS" and "DOTS-CALL-HOME" as application
service tags (Sections 9.3.1 and 9.3.2). It also defines
"signal.udp" (Section 9.3.3), "signal.tcp" (Section 9.3.4), and
"data.tcp" (Section 9.3.5) as application protocol tags. An example
is provided in Figure 8.

In the example below, for domain 'example.net', the resolution
algorithm will result in IP address(es), port, tag and protocol
tuples as follows:

```
example.net.
IN NAPTR 100 10 "" DOTS:signal.udp "" signal.example.net.
IN NAPTR 200 10 "" DOTS:signal.tcp "" signal.example.net.
IN NAPTR 300 10 "" DOTS:data.tcp "" data.example.net.

signal.example.net.
IN NAPTR 100 10 "s" DOTS:signal.udp "" _dots._signal._udp.example.net.
IN NAPTR 200 10 "s" DOTS:signal.tcp "" _dots._signal._tcp.example.net.

data.example.net.
IN NAPTR 100 10 "s" DOTS:data.tcp "" _dots._data._tcp.example.net.

_dots._signal._udp.example.net.
IN SRV   0 0 5000 a.example.net.

_dots._signal._tcp.example.net.
IN SRV   0 0 5001 a.example.net.

_dots._data._tcp.example.net.
IN SRV   0 0 5002 a.example.net.

a.example.net.
IN AAAA  2001:db8::1
```

Order	Protocol	IP address	Port	Tag
1	UDP	2001:db8::1	5000	Signal
2	TCP	2001:db8::1	5001	Signal
3	TCP	2001:db8::1	5002	Data

Figure 8: Sample Example

An example is provided in Figure 9 for the Call Home case.

In the example below, for domain 'example.net', the resolution
algorithm will result in IP address(es), port, tag and protocol
tuples as follows:

```
example.net.
IN NAPTR 100 10 "" DOTS-CALL-HOME:signal.udp "" signal.example.net.
IN NAPTR 200 10 "" DOTS-CALL-HOME:signal.tcp "" signal.example.net.

signal.example.net.
IN NAPTR 100 10 "s" DOTS-CALL-HOME:signal.udp ""
            _dots-call-home._signal._udp.example.net.
IN NAPTR 200 10 "s" DOTS-CALL-HOME:signal.tcp ""
            _dots-call-home._signal._tcp.example.net.

_dots-call-home._signal._udp.example.net.
IN SRV   0 0 6000 b.example.net.

_dots-call-home._signal._tcp.example.net.
IN SRV   0 0 6001 b.example.net.

b.example.net.
IN AAAA  2001:db8::2
```

```
+-------+----------+-------------+------+--------+
| Order | Protocol | IP address  | Port |  Tag   |
+-------+----------+-------------+------+--------+
| 1     | UDP      | 2001:db8::2 | 6000 | Signal |
| 2     | TCP      | 2001:db8::2 | 6001 | Signal |
+-------+----------+-------------+------+--------+
```

Figure 9: Sample Example for DOTS Signal Channel Call Home

If no DOTS-specific S-NAPTR records can be retrieved, the discovery
procedure fails for this domain name (and the corresponding interface
and IP protocol version). If more domain names are known, the
discovery procedure MAY perform the corresponding S-NAPTR lookups
immediately. However, before retrying a lookup that has failed, a
DOTS client MUST wait a time period that is appropriate for the
encountered error (e.g., NXDOMAIN, timeout, etc.).

7. DNS Service Discovery

DNS-based Service Discovery (DNS-SD) [RFC6763] provides generic
solutions for discovering services. DNS-SD defines a set of naming
rules for certain DNS record types that they use for advertising and
discovering services.

Section 4.1 of [RFC6763] specifies that a service instance name in
DNS-SD has the following structure:

<Instance> . <Service> . <Domain>

The <Domain> portion specifies the DNS sub-domain where the service
instance is registered. It may be "local.", indicating the mDNS
local domain, or it may be a conventional domain name such as
"example.com.".

The <Service> portion of the DOTS service instance name MUST be
"_dots._signal._udp" or "_dots._signal._tcp" or "_dots._data._tcp" or
"_dots-call-home._signal._udp" or "_dots-call-home._signal._tcp".

8. Security Considerations

DOTS-related security considerations are discussed in Section 4 of
[I-D.ietf-dots-architecture] is to be considered. DOTS agents must
authenticate each other using (D)TLS before a DOTS session is
considered valid according to the [I-D.ietf-dots-signal-channel].

8.1. DHCP

The security considerations in [RFC2131] and [RFC8415] are to be
considered.

8.2. Service Resolution

The primary attack against the methods described in Section 6 is one
that would lead to impersonation of a DOTS server. An attacker could
attempt to compromise the S-NAPTR resolution. The use of mutual
authentication makes it difficult to redirect a DOTS client to an
illegitimate DOTS server.

8.3. DNS Service Discovery

Since DNS-SD is just a specification for how to name and use records
in the existing DNS system, it has no specific additional security
requirements over and above those that already apply to DNS queries
and DNS updates. For DNS queries, DNS Security Extensions (DNSSEC)
[RFC4033] SHOULD be used where the authenticity of information is
important. For DNS updates, secure updates [RFC2136][RFC3007] SHOULD
generally be used to control which clients have permission to update
DNS records.

9. IANA Considerations

IANA is requested to allocate the SRV service name of "_dots._signal"
for DOTS signal channel over UDP or TCP, and the service name of
"_dots._data" for DOTS data channel over TCP.

9.1. DHCPv6 Option

IANA is requested to assign the following new DHCPv6 Option Code in
the registry maintained in: http://www.iana.org/assignments/
dhcpv6-parameters.

```
Value   Description              Client ORO    Singleton Option
TBD1    OPTION_V6_DOTS_RI        Yes           Yes
TBD2    OPTION_V6_DOTS_ADDRESS   Yes           No
```

9.2. DHCPv4 Option

IANA is requested to assign the following new DHCPv4 Option Code in
the registry maintained in: http://www.iana.org/assignments/bootp-
dhcp-parameters/.

```
          Option Name Value Data length  Meaning
---------------------- ----- ------------ ---------------------------
    OPTION_V4_DOTS_RI TBA3  Variable;    Includes the name of the
                            the maximum  DOTS server.
                            length is
                            255 octets.
 OPTION_V4_DOTS_ADDRESS TBA4 Variable;   Includes one or multiple
                            the minimum  lists of DOTS IP addresses;
                            length is 5. each list is treated as a
                                         separate DOTS server.
```

9.3. Application Service & Application Protocol Tags

This document requests IANA to make the following allocations from
the registry available at: https://www.iana.org/assignments/s-naptr-
parameters/s-naptr-parameters.xhtml.

9.3.1. DOTS Application Service Tag Registration

o Application Protocol Tag: DOTS

o Intended Usage: See Section 6

o Security Considerations: See Section 8

o Contact Information: <one of the authors>

9.3.2. DOTS Call Home Application Service Tag Registration

o Application Protocol Tag: DOTS-CALL-HOME

o Intended Usage: See Section 6

o Security Considerations: See Section 8

o Contact Information: <one of the authors>

9.3.3. signal.udp Application Protocol Tag Registration

o Application Protocol Tag: signal.udp

o Intended Usage: See Section 6

o Security Considerations: See Section 8

o Contact Information: <one of the authors>

9.3.4. signal.tcp Application Protocol Tag Registration

o Application Protocol Tag: signal.tcp

o Intended Usage: See Section 6

o Security Considerations: See Section 8

o Contact Information: <one of the authors>

9.3.5. data.tcp Application Protocol Tag Registration

o Application Protocol Tag: data.tcp

o Intended Usage: See Section 6

o Security Considerations: See Section 8

o Contact Information: <one of the authors>

10. Contributors

Prashanth Patil
Cisco Systems, Inc.

Email: praspati@cisco.com

11. Acknowledgements

Thanks to Brian Carpenter for the review of the BRSKI text.

Many thanks to Russ White for the review, comments, and text
contribution.

Thanks for Dan Wing and Pei Wei for the review and comments.

Thanks to Bernie Volz for the review of the DHCP section.

12. References

12.1. Normative References

[I-D.ietf-dots-signal-channel]
          K, R., Boucadair, M., Patil, P., Mortensen, A., and N.
          Teague, "Distributed Denial-of-Service Open Threat
          Signaling (DOTS) Signal Channel Specification", draft-
          ietf-dots-signal-channel-34 (work in progress), May 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119,
          DOI 10.17487/RFC2119, March 1997,
          <https://www.rfc-editor.org/info/rfc2119>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
          RFC 2131, DOI 10.17487/RFC2131, March 1997,
          <https://www.rfc-editor.org/info/rfc2131>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor
          Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997,
          <https://www.rfc-editor.org/info/rfc2132>.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the
          Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396,
          DOI 10.17487/RFC3396, November 2002,
          <https://www.rfc-editor.org/info/rfc3396>.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application
          Service Location Using SRV RRs and the Dynamic Delegation
          Discovery Service (DDDS)", RFC 3958, DOI 10.17487/RFC3958,
          January 2005, <https://www.rfc-editor.org/info/rfc3958>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing
          Architecture", RFC 4291, DOI 10.17487/RFC4291, February
          2006, <https://www.rfc-editor.org/info/rfc4291>.

   [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service
              Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013,
              <https://www.rfc-editor.org/info/rfc6763>.

   [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman,
              "Special-Purpose IP Address Registries", BCP 153,
              RFC 6890, DOI 10.17487/RFC6890, April 2013,
              <https://www.rfc-editor.org/info/rfc6890>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A.,
              Richardson, M., Jiang, S., Lemon, T., and T. Winters,
              "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)",
              RFC 8415, DOI 10.17487/RFC8415, November 2018,
              <https://www.rfc-editor.org/info/rfc8415>.

12.2. Informative References

   [I-D.ietf-anima-bootstrapping-keyinfra]
              Pritikin, M., Richardson, M., Behringer, M., Bjarnason,
              S., and K. Watsen, "Bootstrapping Remote Secure Key
              Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-
              keyinfra-20 (work in progress), May 2019.

   [I-D.ietf-dots-architecture]
              Mortensen, A., K, R., Andreasen, F., Teague, N., and R.
              Compton, "Distributed-Denial-of-Service Open Threat
              Signaling (DOTS) Architecture", draft-ietf-dots-
              architecture-14 (work in progress), May 2019.

   [I-D.ietf-dots-data-channel]
              Boucadair, M. and R. K, "Distributed Denial-of-Service
              Open Threat Signaling (DOTS) Data Channel Specification",
              draft-ietf-dots-data-channel-29 (work in progress), May
              2019.

   [I-D.ietf-dots-multihoming]
              Boucadair, M. and R. K, "Multi-homing Deployment
              Considerations for Distributed-Denial-of-Service Open
              Threat Signaling (DOTS)", draft-ietf-dots-multihoming-01
              (work in progress), January 2019.

   [I-D.ietf-dots-signal-call-home]
              K, R., Boucadair, M., and J. Shallow, "Distributed Denial-
              of-Service Open Threat Signaling (DOTS) Signal Channel
              Call Home", draft-ietf-dots-signal-call-home-01 (work in
              progress), April 2019.

   [I-D.ietf-dots-use-cases]
              Dobbins, R., Migault, D., Fouant, S., Moskowitz, R.,
              Teague, N., Xia, L., and K. Nishizuka, "Use cases for DDoS
              Open Threat Signaling", draft-ietf-dots-use-cases-17 (work
              in progress), January 2019.

   [I-D.ietf-netconf-zerotouch]
              Watsen, K., Abrahamsson, M., and I. Farrer, "Secure Zero
              Touch Provisioning (SZTP)", draft-ietf-netconf-
              zerotouch-29 (work in progress), January 2019.

   [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound,
              "Dynamic Updates in the Domain Name System (DNS UPDATE)",
              RFC 2136, DOI 10.17487/RFC2136, April 1997,
              <https://www.rfc-editor.org/info/rfc2136>.

   [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic
              Update", RFC 3007, DOI 10.17487/RFC3007, November 2000,
              <https://www.rfc-editor.org/info/rfc3007>.

   [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "DNS Security Introduction and Requirements",
              RFC 4033, DOI 10.17487/RFC4033, March 2005,
              <https://www.rfc-editor.org/info/rfc4033>.

   [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and
              Verification of Domain-Based Application Service Identity
              within Internet Public Key Infrastructure Using X.509
              (PKIX) Certificates in the Context of Transport Layer
              Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March
              2011, <https://www.rfc-editor.org/info/rfc6125>.

   [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and
              S. Krishnan, "Guidelines for Creating New DHCPv6 Options",
              BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014,
              <https://www.rfc-editor.org/info/rfc7227>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: TirumaleswarReddy_Konda@McAfee.com