DOTS
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2021

M. Boucadair
Orange
T. Reddy
McAfee

October 21, 2020

# Distributed-Denial-of-Service Open Threat Signaling (DOTS) Agent Discovery
# draft-ietf-dots-server-discovery-13

## Abstract

This document specifies mechanisms to configure Distributed Denial of Service Open Threat Signaling (DOTS) clients with their DOTS servers. The discovery procedure also covers the DOTS Signal Channel Call Home. Knowing the appropriate DOTS server for a given location can be useful to engage mitigation actions even in cases where the DOTS client cannot localize the attack, but only knows that some resources are under attack and that help is needed.

## Status of This Memo

## Copyright Notice

Table of Contents

## 1. Introduction

DDoS Open Threat Signaling (DOTS) [RFC8811] 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, the DOTS signal channel protocol [RFC8782] is meant to carry requests for DDoS attack mitigation.  This approach allows to reduce the impact of an attack and leads 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 named filtering rules by means of the DOTS data channel protocol [RFC8782].

The basic high-level DOTS architecture is illustrated in Figure 1.

```
                  +-----------+            +-------------+
                  | Mitigator | ~~~~~~~~~~ | DOTS Server |
                  +-----------+            +------+------+
                                                  |
                                                  |
                                                  |
                  +---------------+        +------+------+
                  | Attack Target | ~~~~~~ | DOTS Client |
                  +---------------+        +-------------+
```

Figure 1: Basic DOTS Architecture

[RFC8811] 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 be 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].  The basic high-level DOTS Call Home architecture is illustrated in Figure 2.

```
                +---------------+        +-------------+
                | Alert/DMS/    | ~~~~~~ |  Call Home  |
                | Peer DMS/...  |        | DOTS client |
                +---------------+        +------+------+
                                                |
                                                |
                                                |
                +---------------+        +------+------+
                |    Attack     | ~~~~~~ |  Call Home  |
                |   Source(s)   |        | DOTS server |
                +---------------+        +-------------+
```

Figure 2: Basic DOTS Signal Channel Call Home Functional Architecture

A DOTS agent may be used to establish base DOTS channels, DOTS Call Home, or both. This specification accommodates all these deployment cases.

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

This document assumes that security credentials to authenticate DOTS server(s) are pre-provisioned to a DOTS client using a mechanism such as (but not limited to) those discussed in [RFC8572] or [I-D.ietf-anima-bootstrapping-keyinfra]. DOTS clients use those credentials for authentication purposes following the rules documented in [RFC8782].

## 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 [RFC8811], [RFC3958], and [I-D.ietf-dots-signal-call-home].

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

"Peer DOTS agent" refers to the peer DOTS server (base DOTS operation) or to a peer Call Home 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:

- Many of the 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 existing mechanisms such as discovery using service resolution or DHCP to provision the CPE acting as a DOTS client with the DOTS server(s).

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

- 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 existing protocol behaviors that do not require specific features 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.

- 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).

- 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.

- 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:

- Dynamic discovery using DHCP (Section 5).

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

- DNS Service Discovery (Section 7).

## 4. Unified DOTS Discovery Procedure

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 the following ways to discover DOTS servers in order to facilitate the deployment of DOTS in large scale environments. For example, a CPE will support the first two mechanisms, a host within a LAN will support the last two mechanisms, or an application server will support a local configuration. More examples are discussed in Section 3:

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 IP addresses are configured, a reference identifier must also be configured for authentication purposes.

   - Automatic configuration (e.g., DHCP): 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 suitable DNS 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/IPv6), the DOTS server discovery procedure has to be performed for each interface/address-family combination. A DOTS 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 interface/address-family combinations.

This procedure is also followed by a Call Home DOTS server to discover its Call Home DOTS client in the context of [I-D.ietf-dots-signal-call-home].

The discovery method is reiterated by a DOTS agent upon the following events:

o Expiry of a validity timer (e.g., DHCP lease, DNS TTL) associated with a discovered DOTS agent.

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

o Attachment to a new network.

## 5. DHCP Options for DOTS Agent Discovery

As reported in Section 1.7.2 of [RFC6125]:

> "Some 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 also supplying a name for PKIX-based authentication purposes.

Given that DOTS gateways can be involved in a DOTS session, a peer DOTS agent can be reachable using a link-local address. Such addresses can also be discovered using the options defined in Section 5.1.

The list of the IP addresses returned by DHCP servers is typically used to feed the DOTS server selection procedure including when DOTS agents are provided with primary and backup IP addresses of their peer DOTS agents. An example of DOTS server selection procedure is specified in Section 4.3 of [RFC8782].

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.

The design allows terminating the base DOTS channels and DOTS Call Home on the same or distinct peer DOTS agents. If distinct peer DOTS agents are deployed, the DHCP option can return, for example, a list of IP addresses to a requesting DOTS agent. This list includes the IP address to be used for the base DOTS channels and the IP address for the DOTS Call Home. The DOTS client (or Call Home DOTS server) will then use the address selection procedure specified in Section 4.3 of [RFC8782] to identify the IP address of the peer DOTS server (or Call Home DOTS client). For example:

Let's consider that the DOTS server is reachable at 2001:db8:122:300::1 while the Call Home DOTS client is reachable at 2001:db8:122:300::2. The DHCP server will then return one DOTS reference identifier and a list that includes both 2001:db8:122:300::1 and 2001:db8:122:300::2 to a requesting DHCP client. That list is passed to the DOTS client (or Call Home DOTS

server) which will try to establish connections to the addresses of that list and destination port number 4646 (or the Call Home port number). As a result, the DOTS client (or Call Home DOTS server) will select 2001:db8:122:300::1 (or 2001:db8:122:300::2) as a DOTS server (or Call Home DOTS client).

## 5.1. DHCPv6 DOTS Options

### 5.1.1. Format of DOTS Reference Identifier Option

The DHCPv6 DOTS Reference Identifier option is used to configure a name of the DOTS server (or the name of the Call Home DOTS client). The format of this option is shown in Figure 3.

```
     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 3: DHCPv6 DOTS Reference Identifier Option

The fields of the option shown in Figure 3 are as follows:

- o  Option-code: OPTION_V6_DOTS_RI (TBA1, see Section 9.2)
- o  Option-length: Length of the dots-agent-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 4. 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 4: An example of the dots-agent-name Encoding

### 5.1.2. Format of DOTS Address Option

The DHCPv6 DOTS Address option can be used to configure a list of IPv6 addresses of a DOTS server (or a Call Home DOTS client). The format of this option is shown in Figure 5.

```
     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 5: DHCPv6 DOTS Address Option

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

- Option-code: OPTION_V6_DOTS_ADDRESS (TBA2, see Section 9.2)
- Option-length: Length of the 'DOTS ipv6-address(es)' field in octets. MUST be a multiple of 16.
- DOTS ipv6-address(es): 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.

### 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.

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

If the DHCP client receives and validates both OPTION_V6_DOTS_RI and OPTION_V6_DOTS_ADDRESS, the content of OPTION_V6_DOTS_RI is used as the reference identifier for authentication purposes (e.g., PKIX [RFC6125]), while the valid 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 an underlying resolution library in the presence of valid OPTION_V6_DOTS_ADDRESS in a response.

If the DHCP client receives OPTION_V6_DOTS_RI only, but OPTION_V6_DOTS_RI 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_V6_DOTS_ADDRESS only, the address(es) included in OPTION_V6_DOTS_ADDRESS are used to reach the peer DOTS agent.  In addition, these addresses can be used as identifiers for authentication.

## 5.2.  DHCPv4 DOTS Options

### 5.2.1.  Format of DOTS Reference Identifier Option

The DHCPv4 [RFC2132] DOTS Reference Identifier option is used to configure a name of the peer DOTS agent.  The format of this option is illustrated in Figure 6.

```
          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 6: DHCPv4 DOTS Reference Identifier Option

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

- o Code: OPTION_V4_DOTS_RI (TBA3, see Section 9.3).
- o Length: Includes the length of the "Peer DOTS agent name" field in 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 Address 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 7.

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

Figure 7: DHCPv4 DOTS Address Option

The fields of the option shown in Figure 7 are as follows:

- o Code: OPTION_V4_DOTS_ADDRESS (TBA4, see Section 9.3).
- o Length: is set to 4*N, where N is the number of IPv4 addresses included in the option.
- o DOTS IPv4 Address(es): Contains one or more IPv4 addresses of the peer DOTS agent to be used by a DOTS agent.

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 option, it MUST use only the first instance of that option.

The DHCP client MUST silently discard multicast and host loopback addresses [RFC6890] conveyed in OPTION_V4_DOTS_ADDRESS.

If the DHCP client receives and validates both OPTION_V4_DOTS_RI and OPTION_V4_DOTS_ADDRESS, the content of OPTION_V4_DOTS_RI is used as the reference identifier for authentication purposes (e.g., PKIX [RFC6125]), while the valid 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 valid 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 are used to reach the peer DOTS server. In addition, these addresses can be used as identifiers for authentication.

## 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 agent 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 [RFC3958]. Further DNS lookups may be necessary to determine the peer DOTS agent IP address(es).

Once the DOTS agent has retrieved its DNS domain or discovered the peer DOTS agent name that needs to be resolved, an S-NAPTR lookup with the appropriate application service and the desired protocol tag is made to obtain information necessary to connect to the authoritative peer DOTS agent within the given domain.

This specification defines 'DOTS' and 'DOTS-CALL-HOME' as application service tags (Sections 9.4.1 and 9.4.2). It also defines "signal.udp" (Section 9.4.3), "signal.tcp" (Section 9.4.4), and "data.tcp" (Section 9.4.5) 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 listed in Table 1.

```
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
```

Figure 8: Example of Discovery of DOTS Servers using Service Resolution

| 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 |

Table 1: Resolution Results

An example is provided in Figure 9 for the Call Home case. In this example, the resolution algorithm will result in IP address(es), port, and protocol listed in Table 2 for domain 'example.net'.

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

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

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

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

Figure 9: Example of Discovery of DOTS Call Home Client using Service Resolution

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

Table 2: Resolution Results (Call Home)

Note that customized port numbers are used for the DOTS signal channel, DOTS data channel, and DOTS signal channel call home in the examples shown in Figures 8 and 9 for illustration purposes. If default port numbers are used in a deployment, the discovery

procedure will return 4646 (DOTS signal channel) and 443 (DOTS data channel) as DOTS service port numbers.

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._udp" or "_dots-call-home._tcp".

This document does not define any keys; the TXT record of a DNS-SD service is thus empty (Section 6 of [RFC6763]).

Figure 10 depicts an excerpt of the DNS zone configuration file listing record examples to discover two DOTS signal channel servers. In this example, only UDP is supported as transport for the establishment of the DOTS signal channel.

```
_dots-signal._udp.example.net. PTR  a._dots-signal._udp.example.net.
_dots-signal._udp.example.net. PTR  b._dots-signal._udp.example.net.
a._dots-signal._udp.example.net. SRV 0 0 4646 a.example.net.
b._dots-signal._udp.example.net. SRV 0 0 4646 b.example.net.
a._dots-signal._udp.example.net. TXT ""
b._dots-signal._udp.example.net. TXT ""
```

Figure 10: An Example of DNS-SD Records for the UDP DOTS Signal Channel involving Two Servers with the Same Priority.

## 8. Security Considerations

DOTS-related security considerations are discussed in Section 4 of [RFC8811]. As a reminder, DOTS agents must authenticate each other using (D)TLS before a DOTS session is considered valid according to the [RFC8782].

An attacker may block some protocol messages (e.g., DHCP) to force the client to use a discovery mechanism with a lower priority. The security implications of such attack are those inherent to the fallback discovery mechanism discussed in the following subsections.

The results of the discovery procedure are a function of the interface/address family. Contacting a discovered DOTS server via an interface to which it is not bound may exacerbate the delay required to establish a DOTS channel. Moreover, such behavior may reveal that a DOTS service is enabled by a DOTS client domain and exposes the identity of the DOTS service provider (that can be inferred from the name and the destination IP address) to external networks.

Security considerations related to how security credentials to authenticate DOTS server(s) are provisioned to a DOTS client are those inherent to the mechanism used for that purpose (see for example, [RFC8572]).

### 8.1. DHCP

The security considerations in [RFC2131] and [RFC8415] are to be considered. In particular, issues related to rogue DHCP servers and means to mitigate many of these attacks are discussed in Section 22 of [RFC8415].

An attacker can get a domain name, domain-validated public certificate from a CA, and host a DOTS agent. An active attacker can then spoof DHCP responses to include the attacker's DOTS agent. Such an attacker can also launch other attacks as discussed in Section 22 of [RFC8415]. In addition to the mitigations listed in Section 22 of [RFC8415], a DOTS agent may be pre-configured with a list of trusted

DOTS domain names. If such a list is pre-configured, a DOTS agent will accept a DHCP-discovered name if it matches a name in that list. Also, the DOTS agent has to check that the 'DNS-ID' identifier type within subjectAltName in the server certificate matches a pre-configured name. If the DOTS agent is instructed to trust subdomains of the names in that list as well, a DOTS agent will also accept a DHCP-discovered name if the left-most label of the discovered name is matching a name in the pre-configured list.

Relying on an underlying resolution library to resolve a supplied reference identifier has similar security issues as those discussed in Section 8.2 (e.g., an active attacker may modify DNS messages used to resolve the supplied reference identifier and point the client to an attacker server).

Supplying both an IP address and the reference identifier makes it easier to use a mis-issued certificate.

### 8.2. Service Resolution

The primary attack against the methods described in Section 6 is one that would lead to impersonation of a peer DOTS agent. An attacker could attempt to compromise the S-NAPTR resolution.

The DOTS client (or a Call Home DOTS server) constructs one reference identifier for the DOTS server (or a Call Home DOTS client) based on the domain name which is used for S-NAPTR lookup: DNS-ID. If the reference identifier is found (as described in Section 6 of [RFC6125]) in the PKIX certificate's subjectAltName extension, the DOTS client should accept the certificate for the server.

DNS Security Extensions (DNSSEC) [RFC4033] uses cryptographic keys and digital signatures to provide authentication of DNS data. The information that is retrieved from the S-NAPTR lookup and that is validated using DNSSEC is thereby proved to be the authoritative data.

### 8.3. DNS Service Discovery

Since DNS-SD is 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, DNSSEC 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

### 9.1. The Service Name and Transport Protocol Port Number Registry

IANA is requested to allocate the following service names from the registry available at: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

| | |
|---|---|
| Service Name: | dots-data |
| Port Number: | N/A |
| Transport Protocol(s): | TCP |
| Description: | DOTS Data Channel Protocol<br>The service name is used to construct the SRV service name "_dots-data._tcp" for discovering DOTS servers used to establish DOTS data channel. |
| Assignee: | IESG <iesg@ietf.org> |
| Contact: | IETF Chair <chair@ietf.org> |
| Reference: | [ThisDocument] |

| | |
|---|---|
| Service Name: | dots-call-home |
| Transport Protocol(s): | TCP/UDP |
| Description: | DOTS Signal Channel Call Home Protocol.<br>The service name is used to construct the SRV service names "_dots-call-home._udp" and "_dots-call-home._tcp" for discovering Call Home DOTS clients used to establish DOTS signal channel call home. |
| Assignee: | IESG <iesg@ietf.org> |
| Contact: | IETF Chair <chair@ietf.org> |
| Reference: | [ThisDocument] |

IANA is requested to update the following entry from the registry available at: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

| | |
|---|---|
| Service Name: | dots-signal |
| Port Number: | 4646 |
| Transport Protocol(s): | TCP/UDP |
| Description: | DOTS Signal Channel Protocol.<br>The service name is used to construct the SRV service names "_dots-signal._udp" and "_dots-signal._tcp" for discovering DOTS servers used to establish DOTS signal channel. |
| Assignee: | IESG <iesg@ietf.org> |
| Contact: | IETF Chair <chair@ietf.org> |
| Reference: | [RFC8782][ThisDocument] |

### 9.2. DHCPv6 Options

IANA is requested to assign the following new DHCPv6 Option Codes in the registry maintained in: https://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml#dhcpv6-parameters-2.

| Value | Description | Client ORO | Singleton Option |
|---|---|---|---|
| TBA1 | OPTION_V6_DOTS_RI | Yes | Yes |
| TBA2 | OPTION_V6_DOTS_ADDRESS | Yes | Yes |

### 9.3. DHCPv4 Options

IANA is requested to assign the following new DHCPv4 Option Codes in the registry maintained in: https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#options.

| Name | Tag | Data Length | Meaning | Reference |
|---|---|---|---|---|
| OPTION_V4_DOTS_RI | TBA3 | N | The name of the peer DOTS agent. | [ThisDocument] |
| OPTION_V4_DOTS_ADDRESS | TBA4 | N (the minimal length is 4) | N/4 IPv4 addresses of peer DOTS agent(s). | [ThisDocument] |

### 9.4. Application Service & Application Protocol Tags

This document requests IANA to make the following allocations from the registries available at: https://www.iana.org/assignments/s-naptr-parameters/s-naptr-parameters.xhtml#s-naptr-parameters-1 for Application Service Tags and https://www.iana.org/assignments/s-naptr-parameters/s-naptr-parameters.xhtml#s-naptr-parameters-2 for Application Protocol Tags.

#### 9.4.1. DOTS Application Service Tag Registration

- Application Service Tag: DOTS

- Intended Usage: See Section 6

- Security Considerations: See Section 8

- Interoperability considerations: None

- Relevant publications: This document

### 9.4.2. DOTS Call Home Application Service Tag Registration

- Application Service Tag: DOTS-CALL-HOME
- Intended Usage: See Section 6
- Security Considerations: See Section 8
- Interoperability considerations: None
- Relevant publications: This document

### 9.4.3. signal.udp Application Protocol Tag Registration

- Application Protocol Tag: signal.udp
- Intended Usage: See Section 6
- Security Considerations: See Section 8
- Interoperability considerations: None
- Relevant publications: This document

### 9.4.4. signal.tcp Application Protocol Tag Registration

- Application Protocol Tag: signal.tcp
- Intended Usage: See Section 6
- Security Considerations: See Section 8
- Interoperability considerations: None
- Relevant publications: This document

### 9.4.5. data.tcp Application Protocol Tag Registration

- Application Protocol Tag: data.tcp
- Intended Usage: See Section 6
- Security Considerations: See Section 8
- Interoperability considerations: None
- Relevant publications: This document

## 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, Pei Wei, Valery Smyslov, and Jon Shallow for the review and comments.

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

Many thanks to Benjamin Kaduk for the detailed AD review.

Thanks to Zhen Cao, Kyle Rose, Nagendra Nainar, and Peter Yee for the directorate reviews.

## 12. References

### 12.1. Normative References

[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., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-44 (work in progress), September 2020.

[I-D.ietf-dots-multihoming] Boucadair, M., Reddy.K, T., and W. Pan, "Multi-homing Deployment Considerations for Distributed-Denial-of-Service Open Threat Signaling (DOTS)", draft-ietf-dots-multihoming-04 (work in progress), May 2020.

[I-D.ietf-dots-signal-call-home] Reddy.K, T., Boucadair, M., and J. Shallow, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Call Home", draft-ietf-dots-signal-call-home-09 (work in progress), September 2020.

[I-D.ietf-dots-use-cases]
Dobbins, R., Migault, D., Moskowitz, R., Teague, N., Xia, L., and K. Nishizuka, "Use cases for DDoS Open Threat Signaling", draft-ietf-dots-use-cases-25 (work in progress), July 2020.

[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>.

[RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <https://www.rfc-editor.org/info/rfc8572>.

[RFC8782] Reddy.K, T., Ed., Boucadair, M., Ed., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", RFC 8782, DOI 10.17487/RFC8782, May 2020, <https://www.rfc-editor.org/info/rfc8782>.

[RFC8783] Boucadair, M., Ed. and T. Reddy.K, Ed., "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", RFC 8783, DOI 10.17487/RFC8783, May 2020, <https://www.rfc-editor.org/info/rfc8783>.

[RFC8811] Mortensen, A., Ed., Reddy.K, T., Ed., Andreasen, F., Teague, N., and R. Compton, "DDoS Open Threat Signaling (DOTS) Architecture", RFC 8811, DOI 10.17487/RFC8811, August 2020, <https://www.rfc-editor.org/info/rfc8811>.

## 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