Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2018

M. Boucadair
Orange
T. Reddy
McAfee
P. Patil
Cisco

October 18, 2017

Distributed-Denial-of-Service Open Threat Signaling (DOTS) Server Discovery
draft-boucadair-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 nodes with DOTS servers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In many deployments, it may not be possible for a network to determine the cause for a distributed Denial-of-Service (DoS) attack [RFC4732], but instead just realize that some resources seem to be under attack. To fill that gap, the IETF is specifying an architecture, called DDoS Open Threat Signaling (DOTS) [I-D.ietf-dots-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 protocol is meant to carry requests for DDoS attack mitigation, thereby reducing the impact of an attack and leading to more efficient defensive actions. [I-D.ietf-dots-use-cases] identifies a set of scenarios for DOTS.

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. The logic for connecting to one or multiple IP addresses is out of scope of this document.

This document specifies methods for DOTS clients to discover their DOTS server(s). The rationale for specifying multiple discovery mechanisms is discussed in Section 4.

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

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

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

This document makes use of the following terms:

- DDoS: A distributed Denial-of-Service attack, in which traffic originating from multiple sources are directed at a target on a network. DDoS attacks are intended to cause a negative impact on the availability of servers, services, applications, and/or other functionality of an attack target.
- DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- DHCP client denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers.
- DHCP server refers to a node that responds to requests from DHCP clients.
- DOTS client: A DOTS-aware software module responsible for requesting attack response coordination with other DOTS-aware elements.
- DOTS server: A DOTS-aware software module handling and responding to messages from DOTS clients. The DOTS server should enable mitigation on behalf of the DOTS client, if requested, by communicating the DOTS client's request to the mitigator and returning selected mitigator feedback to the requesting DOTS client. A DOTS server may also be a mitigator.
- DOTS gateway: A DOTS-aware software module that is logically equivalent to a DOTS client back-to-back with a DOTS server.

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

4. 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 (Table 1). Concretely:

- 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 provision the preferred DOTS signal channel transport protocol between the DOTS client and the DOTS server and allows the DOTS client to discover this preference.

- Resolving a DOTS server domain name offered by the 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.

- The upstream network provider is not the DDoS mitigation provider for some of these use cases. The use of anycast is not appropriate for this use case, in particular. 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). Automatic 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 forking requests received from DOTS clients to upstream DOTS servers or for selecting the appropriate DOTS server. Particularly, the use of anycast may simplify the operations within the enterprise network to discover a DOTS gateway, if the enterprise network is single-homed.

- 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 or anycast to provision the CPE acting as a DOTS client with the DOTS server(s).

Use Case	Requires a CPE	The Network Provider is also the DDoS Mitigation Provider
End-customer with single or multiple upstream transit provider(s) offering DDoS mitigation services	Yes (Intelligent DDoS mitigation system (IDMS) acting as a DOTS client may be co-located on the CPE)	Yes
End-customer with an overlay DDoS mitigation managed security service provider (MSSP)	Yes (DDOS Detector acting as a DOTS client may be co-located on the CPE)	No
End-customer operating an application or service with an integrated DOTS client	Yes (CPE may act as a DOTS gateway)	Yes/No
End-customer operating a CPE network infrastructure device with an integrated DOTS client	Yes (CPE acts as a DOTS client)	Yes
Suppression of outbound DDoS traffic originating from a consumer broadband access network	Yes (CPE acts as a DOTS server)	Yes
DDoS Orchestration	No	N/A

Table 1: Summary of DOTS Use Cases

Consequently, this document describes the following mechanisms for discovery:

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

- DNS Service Discovery.

- Discovery using DHCP Options.

- A mechanism based on anycast address for DOTS usage.

5. 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 server 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 DNS discovery, while others prefer DNS 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, 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 four mechanisms below to determine a DOTS server list. All DOTS clients SHOULD implement all four, 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' 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 9.

2. Service Resolution : The DOTS client attempts to discover DOTS server name(s) using service resolution, as specified in Section 7.

3. DNS SD: DNS Service Discovery. The DOTS client attempts to discover DOTS server name(s) using DNS service discovery, as specified in Section 8.

4. Anycast : Send DOTS request to establish a DOTS session with the assigned DOTS server anycast address for each combination of interface and address family.

Some of these mechanisms imply the use of DNS to resolve the IP address of the DOTS server, while others imply the 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.

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.

6. Resolution

Once the DOTS client has retrieved client's DNS domain or discovered the DOTS server name that needs to be resolved, 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" as an application service tag (Section 12.3.1) and "signal.udp" (Section 12.3.2), "signal.tcp" (Section 12.3.3), and "data.tcp" (Section 12.3.4) as application protocol tags.

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

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. 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.

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).

7.1. Retrieving Domain Name

A DOTS client has to determine the domain in which it is located. The following section describes the means to obtain the domain name from DHCP. Other means of retrieving domain names may be used, which are outside the scope of this document, e.g., local configuration.

Implementations MAY allow the user to specify a default name that is used, if no specific name has been configured.

7.1.1. DHCP

DHCP can be used to determine the domain name related to an interface's point of network attachment. Network operators may provide the domain name to be used for service discovery within an access network using DHCP. Sections 3.2 and 3.3 of [RFC5986] define DHCP IPv4 and IPv6 access network domain name options, OPTION_V4_ACCESS_DOMAIN and OPTION_V6_ACCESS_DOMAIN respectively, to identify a domain name that is suitable for service discovery within the access network.

For IPv4, the discovery procedure MUST request the access network domain name option in a Parameter Request List option, as described in [RFC2131]. [RFC2132] defines the DHCP IPv4 domain name option; while this option is less suitable, a client MAY request for it if the access network domain name defined in [RFC5986] is not available.

For IPv6, the discovery procedure MUST request for the access network domain name option in an Options Request Option (ORO) within an Information-request message, as described in [RFC3315].

If neither option can be retrieved the procedure fails for this interface. If a result can be retrieved it will be used as an input for S-NAPTR resolution discussed in Section 6.

8. DNS Service Discovery

DNS-based Service Discovery (DNS-SD) [RFC6763] and Multicast DNS (mDNS) [RFC6762] provide generic solutions for discovering services. DNS-SD/mDNS define a set of naming rules for certain DNS record types that they use for advertising and discovering services.

8.1. DNS-SD

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".

8.2. mDNS

A DOTS client can proactively discover DOTS servers being advertised in the site by multicasting a PTR query to one or all of the following:

- "_dots._signal._udp.local."
- "_dots._signal._tcp.local."
- "_dots._data._tcp.local."

A DOTS server can send out gratuitous multicast DNS answer packets whenever it starts up, wakes from sleep, or detects a change in network configuration. DOTS clients receive these gratuitous packets and cache information contained in it.

9. 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.

9.1. DHCPv6 DOTS Options

9.1.1. Format of DOTS Reference Identifier Option

The DHCPv6 DOTS option is used to configure a name of the DOTS server. 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            |         Option-length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                      dots-server-name (FQDN)                  |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: DHCPv6 DOTS Reference Identifier option

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

- Option-code: OPTION_V6_DOTS_RI (TBA1, see Section 12.1)
- Option-length: Length of the dots-server-name field in octets.
- dots-server-name: A fully qualified domain name of the DOTS server. This field is formatted as specified in Section 8 of [RFC3315].

An example of the dots-server-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-server-name encoding

9.1.2. Format Format of DOTS Address Option

The DHCPv6 DOTS option can be used to configure a list of IPv6 addresses of a DOTS server. The format of this option is shown in Figure 4.

```
  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            |         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:

- Option-code: OPTION_V6_DOTS_ADDRESS (TBA2, see Section 12.1)
- Option-length: Length of the 'DOTS ipv6-address(es)' field in octets. MUST be a multiple of 16.
- DOTS ipv6-address: Includes one or more IPv6 addresses [RFC4291] of the DOTS server to be used by the DOTS client.

  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 servers to the requesting DHCPv6 client, the DHCPv6 server returns multiple instances of OPTION_V6_DOTS.

9.1.3. DHCPv6 Client Behavior

DHCP clients MAY request options OPTION_V6_DOTS_RI and OPTION_V6_DOTS_ADDRESS, as defined in [RFC3315], Sections 17.1.1, 18.1.1, 18.1.3, 18.1.4, 18.1.5, and 22.7. 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 (resp. 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 DOTS server. 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 [RFC3315]), 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 DOTS server. In addition, these addresses can be used as identifiers for authentication.

9.2. DHCPv4 DOTS Options

9.2.1. Format of DOTS Reference Identifier Option

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

```
          Code  Length   DOTS server name
         +-----+-----+-----+-----+-----+-----+-----+--
         | TBA |  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:

- Code: OPTION_V4_DOTS_RI (TBA3, see Section 12.2);
- Length: Includes the length of the "DOTS server name" field in octets; the maximum length is 255 octets.

- DOTS server name: The domain name of the DOTS server. This field is formatted as specified in Section 8 of [RFC3315].

9.2.2. Format Format of DOTS Address Option

The DHCPv4 DOTS option can be used to configure a list of IPv4 addresses of a DOTS server. 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         |     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:

- Code: OPTION_V4_DOTS_ADDRESS (TBA4, see Section 12.2);
- Length: Length of all included data in octets. The minimum length is 5.
- List-Length: Length of the "List of DOTS IPv4 Addresses" field in octets; MUST be a multiple of 4.
- List of DOTS IPv4 Addresses: Contains one or more IPv4 addresses of the DOTS server to be used by the DOTS client. The format of this field is shown in Figure 7.
- OPTION_V4_DOTS can include multiple lists of DOTS IPv4 addresses; each list is treated separately as it corresponds to a given DOTS server.

  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 is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION_V4_DOTS exceeds the maximum DHCPv4 option size of 255 octets.

9.2.3. DHCPv4 Client Behavior

To discover a DOTS server, 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 (resp. 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 DOTS server. 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 8 of [RFC3315]), 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 DOTS server. In addition, these addresses can be used as identifiers for authentication.

10. Anycast

IP anycast can also be used for DOTS service discovery. A packet sent to an anycast address is delivered to the 'topologically nearest' network interface with the anycast address.

When a DOTS client requires DOTS services, it attempts to establish a signaling session with the assigned anycast address(es) defined in Sections 12.4 and 12.5. A DOTS server, that receives a DOTS request with an anycast address, SHOULD redirect the DOTS client to the appropriate DOTS unicast server(s) using the mechanism described in Section 5.5 of [I-D.ietf-dots-signal-channel], unless it is configured otherwise. Indeed, a DOTS server SHOULD be configurable to maintain all DOTS communications using anycast. DOTS redirect is not made mandatory because the use of anycast is not problematic for some deployment scenarios such as an enterprise network deploying one single DOTS gateway connected to one single network provider.

[I-D.boucadair-dots-multihoming] identifies a set of deployment schemes in which the use of anycast is not recommended.

11. 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.

If the DOTS client is explicitly configured with DOTS server(s) then the DOTS client can also be explicitly configured with credentials to authenticate the DOTS server.

The CPE device acting as a DOTS client MAY use Bootstrapping Remote Secure Key Infrastructures (BRSKI) discussed in [I-D.ietf-anima-bootstrapping-keyinfra] to automatically bootstrap using the vendor installed X.509 certificate, in combination with a domain registrar provided by the upstream transit provider and vendor's authorizing service. The CPE device authenticates to the upstream transit provider using the vendor installed X.509 certificate and the upstream transit provider validates the vendor installed certificate on the CPE device using the Manufacturer Authorized Signing Authority (MASA) service. If authentication is successful then the CPE device can request and get a voucher from the MASA service via the domain registrar. The voucher is signed by the MASA service and includes the upstream transit provider's trust anchor certificate. The CPE device validates the signed voucher using the manufacturer installed trust anchor associated with the vendor's selected MASA service and stores the upstream transit

provider's trust anchor certificate. The CPE device then uses Enrollment over Secure Transport (EST) [RFC7030] for certificate enrollment (Section 3.8 in [I-D.ietf-anima-bootstrapping-keyinfra]). The DOTS client on the CPE device can authenticate to the DOTS server using the certificate provisioned by the EST server and the DOTS client can validate the DOTS server certificate using the upstream transit provider's trust anchor certificate it had received in the voucher.

11.1. DHCP

The security considerations in [RFC2131] and [RFC3315] are to be considered.

11.2. Service Resolution

The primary attack against the methods described in Section 7 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.

11.3. 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.

For mDNS, in addition to what has been described above, a principal security threat is a security threat inherent to IP multicast routing and any application that runs on it. A rogue system can advertise that it is a DOTS server. Discovery of such rogue systems as DOTS servers, in itself, is not a security threat if the DOTS client authenticates the discovered DOTS servers.

11.4. Anycast

Anycast-related security considerations are discussed in [RFC4786] and [RFC7094].

12. 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.

12.1. 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:

Option Name	Value
OPTION_V6_DOTS_RI	TBA1
OPTION_V6_DOTS_ADDRESS	TBA2

12.2. 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; the maximum length is 255 octets.	Includes the name of the DOTS server.
OPTION_V4_DOTS_ADDRESS	TBA4	Variable; the minimum length is 5.	Includes one or multiple lists of DOTS IP addresses; each list is treated as a separate DOTS server.

12.3. 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.

12.3.1. DOTS Application Service Tag Registration

- Application Protocol Tag: DOTS

- Intended Usage: See Section 6

- Security Considerations: See Section 11

- Contact Information: <one of the authors>

12.3.2. signal.udp Application Protocol Tag Registration

- Application Protocol Tag: signal.udp

- Intended Usage: See Section 6

- Security Considerations: See Section 11

- Contact Information: <one of the authors>

12.3.3. signal.tcp Application Protocol Tag Registration

- Application Protocol Tag: signal.tcp

- Intended Usage: See Section 6

- Security Considerations: See Section 11

- Contact Information: <one of the authors>

12.3.4. data.tcp Application Protocol Tag Registration

- Application Protocol Tag: data.tcp

- Intended Usage: See Section 6

- Security Considerations: See Section 11

- Contact Information: <one of the authors>

12.4. IPv4 Anycast

IANA has assigned a single IPv4 address from the 192.0.0.0/24 prefix and registered it in the "IANA IPv4 Special-Purpose Address Registry" [RFC6890].

Attribute	Value
Address Block	TBA
Name	Distributed-Denial-of-Service Open Threat Signaling (DOTS) Anycast
RFC	<this document>
Allocation Date	<date of approval of this document>
Termination Date	N/A
Source	True
Destination	True
Forwardable	True
Global	True
Reserved-by-Protocol	False

12.5. IPv6 Anycast

IANA has assigned a single IPv6 address from the 2001:0000::/23 prefix and registered it in the "IANA IPv6 Special-Purpose Address Registry" [RFC6890].

Attribute	Value
Address Block	TBA
Name	Distributed-Denial-of-Service Open Threat Signaling (DOTS) Anycast
RFC	<this document>
Allocation Date	<date of approval of this document>
Termination Date	N/A
Source	True
Destination	True
Forwardable	True
Global	True
Reserved-by-Protocol	False

13. Acknowledgements

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

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

14. References

14.1. Normative References

[I-D.ietf-dots-architecture]
Mortensen, A., Andreasen, F., Reddy, T., christopher_gray3@cable.comcast.com, c., Compton, R., and N. Teague, "Distributed-Denial-of-Service Open Threat Signaling (DOTS) Architecture", draft-ietf-dots-architecture-04 (work in progress), July 2017.

[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>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <https://www.rfc-editor.org/info/rfc3315>.

[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>.

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, DOI 10.17487/RFC5986, September 2010, <https://www.rfc-editor.org/info/rfc5986>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[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>.

14.2. Informative References

[I-D.boucadair-dots-multihoming]
Boucadair, M. and T. Reddy, "Multi-homing Deployment Considerations for Distributed-Denial-of-Service Open Threat Signaling (DOTS)", draft-boucadair-dots-multihoming-02 (work in progress), October 2017.

[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-08 (work in progress), October 2017.

[I-D.ietf-dots-signal-channel]
Reddy, T., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel", draft-ietf-dots-signal-channel-05 (work in progress), October 2017.

[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-07 (work in progress), July 2017.

[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>.

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <https://www.rfc-editor.org/info/rfc4732>.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <https://www.rfc-editor.org/info/rfc4786>.

[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>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <https://www.rfc-editor.org/info/rfc7094>.

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

Prashanth Patil
Cisco Systems, Inc.

Email: praspati@cisco.com