DOTS
Internet-Draft
Intended status: Informational
Expires: September 20, 2016

A. Mortensen
Arbor Networks, Inc.
F. Andreasen
T. Reddy
Cisco Systems, Inc.
C. Gray
Comcast, Inc.
R. Compton
Charter Communications, Inc.
N. Teague
Verisign, Inc.

March 19, 2016

Distributed-Denial-of-Service (DDoS) Open Threat Signaling Architecture
draft-mortensen-dots-architecture-00

Abstract

This document describes an architecture for establishing and maintaining Distributed Denial of Service (DDoS) Open Threat Signaling (DOTS) within and between networks. The document makes no attempt to suggest protocols or protocol extensions, instead focusing on architectural relationships, components and concepts used in a DOTS deployment.

Status of This Memo

Copyright Notice

Table of Contents

1. Context and Motivation

Signaling the need for help defending against an active distributed denial of service (DDoS) attack requires a common understanding of mechanisms and roles among the parties coordinating attack response. The proposed signaling layer and supplementary messaging is the focus

of DDoS Open Threat Signaling (DOTS). DOTS proposes to standardize a method of coordinating defensive measures among willing peers to mitigate attacks quickly and efficiently.

This document describes an architecture used in establishing, maintaining or terminating a DOTS relationship in a network or between networks. DOTS enables hybrid attack responses, coordinated locally at or near the target of an active attack, as well as closer to attack sources in the network path.

1.1. Terminology

1.1.1. Key Words

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 [RFC2119].

1.1.2. Definition of Terms

This document uses the terms defined in [I-D.ietf-dots-requirements].

1.2. Scope

This document defines an architecture for the proposed DOTS standard in the IETF.

In this architecture, DOTS clients and servers communicate using the signaling mechanism established in the proposed DOTS standard. As a result of signals from a DOTS client, the DOTS server may modify the network path of traffic destined for the attack target or targets, for example by diverting traffic to a scrubbing center. Packets deemed part of an active attack may be dropped.

The architecture presented here is assumed to be applicable across network administrative domains - for example, between an enterprise domain and the domain of a third-party attack scrubbing service - as well as to a single administrative domain. DOTS is generally assumed to be most effective when aiding coordination of attack response between two or more participating network domains, but single domain scenarios are valuable in their own right, as when aggregating intra-domain DOTS client signals for inter-domain coordinated attack response.

1.3. Assumptions

This document makes the following assumptions:

- The network or networks in which DOTS is deployed are assumed to offer the required connectivity between DOTS agents and any intermediary network elements, but the architecture imposes no additional limitations on the form of connectivity.

- Congestion and resource exhaustion are intended outcomes of a DDoS attack [RFC4732]. Some operators may utilize non-impacted paths or networks for DOTS, however, it should be assumed that, in general, conditions will be hostile and that DOTS must be able to function in all circumstances, including when the signaling path is significantly impaired.

- There is no universal DDoS attack scale threshold triggering a coordinated response across network administrative domains. A network domain administrator, or service or application owner may arbitrarily set attack scale threshold triggers, or manually send requests for mitigation.

- The mitigation capacity and/or capability of networks receiving requests for coordinated attack response is opaque to the network sending the request. The entity receiving the DOTS client signal may or may not have sufficient capacity or capability to filter any or all DDoS attack traffic directed at a target.

- DOTS client and server signals, as well as messages sent through the data channel, are sent across any transit networks with the same probability of delivery as any other traffic between the DOTS client network and the DOTS server network. Any encapsulation required for successful delivery is left untouched by transit network elements. DOTS server and DOTS client cannot assume any preferential treatment of DOTS signals.

- The architecture allows for, but does not assume, the presence of Quality of Service (QoS) policy agreements between DOTS-enabled peer networks or local QoS prioritization aimed at ensuring delivery of DOTS messages between DOTS agents. QoS is an operational consideration only, not a functional part of a DOTS architecture.

- There is no assumption that the signal channel and the data channel should terminate on the same DOTS server: they may be loosely coupled.

2. Architecture

DOTS enables a target that is under a Distributed Denial-of-Service (DDoS) attack to signal another entity for help in mitigating the DDoS attack. The basic high-level DOTS architecture is illustrated in Figure 1:

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

Figure 1: Basic DOTS Architecture

A simple example instantiation of the DOTS architecture could be an enterprise as the attack target for a volumetric DDoS attack, and an upstream DDoS mitigation service as the Mitigator. The enterprise (attack target) is connected to the Internet via a link that is getting saturated, and the enterprise suspects it is under DDoS attack. The enterprise has a DOTS client, which obtains information about the DDoS attack, and signals the DOTS server for help in mitigating the attack. The communication may be direct from the DOTS client to the DOTS Server, or it may traverse one or more DOTS Relays, which act as intermediaries. The DOTS Server in turn invokes one or more mitigators, which are tasked with mitigating the actual DDoS attack, and hence aim to suppress the attack traffic while allowing valid traffic to reach the attack target.

The scope of the DOTS specifications is the interfaces between the DOTS client, DOTS server, and DOTS relay. The interfaces to the attack target and the mitigator are out of scope of DOTS. Similarly, the operation of both the attack target and the mitigator are out of scope of DOTS. Thus, DOTS neither specifies how an attack target decides it is under DDoS attack, nor does DOTS specify how a mitigator may actually mitigate such an attack. Indeed, a DOTS client's request for mitigation is advisory in nature, and may not lead to any mitigation at all, depending on the DOTS server entity's

capacity and willingness to mitigate on behalf of the DOTS client's entity.

As illustrated in Figure 2, there are two interfaces between the DOTS Server and the DOTS Client (and possibly the DOTS Relay):

```
  +---------------+                                 +---------------+
  |               | <------- Signal Channel ------> |               |
  |  DOTS Client  |                                 |  DOTS Server  |
  |               | <=======  Data Channel  ======> |               |
  +---------------+                                 +---------------+
```

Figure 2: DOTS Interfaces

The primary purpose of the signal channel is for the DOTS client to ask the DOTS server for help in mitigating an attack, and for the DOTS server to inform the DOTS client about the status of such mitigation. The DOTS client does this by sending a client signal, which contains information about the attack target or targets. The client signal may also include telemetry information about the attack, if the DOTS client has such information available. The DOTS Server in turn sends a server signal to inform the DOTS client of whether it will honor the mitigation request. Assuming it will, the DOTS Server initiates attack mitigation (by means outside of DOTS), and periodically informs the DOTS client about the status of the mitigation. Similarly, the DOTS client periodically informs the DOTS server about the client's status, which at a minimum provides client (attack target) health information, but it may also include telemetry information about the attack as it is now seen by the client. At some point, the DOTS client may decide to terminate the server-side attack mitigation, which it indicates to the DOTS server over the signal channel. A mitigation may also be terminated if a DOTS client-specified mitigation time limit is exceeded; additional considerations around mitigation time limits may be found below. Note that the signal channel may need to operate over a link that is experiencing a DDoS attack and hence is subject to severe packet loss and high latency.

While DOTS is able to request mitigation with just the signal channel, the addition of the DOTS data channel provides for additional and more efficient capabilities; both channels are required in the DOTS architecture. The primary purpose of the data channel is to support DOTS related configuration and policy information exchange between the DOTS client and the DOTS server. Examples of such information include

- o Defining names or aliases for attack targets (resources). Those names can be used in subsequent signal channel exchanges to more efficiently refer to the resources (attack targets) in question.

- o Black-list management, which enables a DOTS client to inform the DOTS server about sources to suppress.

- o White-list management, which enables a DOTS client to inform the DOTS server about sources from which traffic should always be accepted.

- o DOTS client provisioning.

- o Vendor-specific extensions, supplementing or in some other way facilitating mitigation when the mitigator relies on particular proprietary interfaces.

Note that while it is possible to exchange the above information before, during or after a DDoS attack, DOTS requires reliable delivery of the above information and does not provide any special means for ensuring timely delivery of it during an attack. In practice, this means that DOTS entities SHOULD NOT rely on such information being exchanged during a DDoS attack.

2.1. DOTS Operations

The scope of DOTS is focused on the signaling and data exchange between the DOTS client, DOTS server and (possibly) the DOTS relay. DOTS does not prescribe any specific deployment models, however DOTS is designed with some specific requirements around the different DOTS agents and their relationships.

First of all, a DOTS agent belongs to an entity, and that entity has an identity which can be authenticated. DOTS agents communicate with each other over a mutually authenticated signal channel and bulk data channel. However, before they can do so, a service relationship needs to be established between them. The details and means by which this is done is outside the scope of DOTS, however an example would be for an enterprise A (DOTS client) to sign up for DDoS service from provider B (DOTS server). This would establish a (service) relationship between the two that enables enterprise A's DOTS client to establish a signal channel with provider B's DOTS server. A and B will authenticate each other, and B can verify that A is authorized for its service. A and B may each have one or more DOTS relays in front of their DOTS client and DOTS server.

[[EDITOR'S NOTE: we request working group feedback and discussion of considerations of end-to-end signaling and agent authentication/ authorization with relays in the signaling path.]]

From an operational and design point of view, DOTS assumes that the above relationship is established prior to a request for DDoS attack mitigation. In particular, it is assumed that bi-directional communication is possible at this time between the DOTS client and DOTS server. Furthermore, it as assumed that additional service provisioning, configuration and information exchange can be performed by use of the data channel, if operationally required. It is not until this point that the mitigation service is available for use.

Once the mutually authenticated signal channel has been established, it will remain in place. This is done to increase the likelihood that the DOTS client can signal the DOTS server for help when the attack target is being flooded, and similarly raise the probability that DOTS server signals reach the client regardless of inbound link congestion. This does not necessarily imply that the attack target and the DOTS client have to be co-located in the same administrative domain, but it is expected to be a common scenario.

DDoS mitigation service with the help of an upstream mitigator will often involve some form of traffic redirection whereby traffic destined for the attack target is diverted towards the mitigator, e.g. by use of BGP [RFC4271] or DNS [RFC1034]. The mitigator in turn inspects and scrubs the traffic, and forwards the resulting (hopefully non-attack) traffic to the attack target, e.g. via a GRE tunnel. Thus, when a DOTS server receives an attack mitigation request from a DOTS client, it can be viewed as a way of causing traffic redirection for the attack target indicated. Note that DOTS does not consider any authorization aspects around who should be allowed to issue such requests for what attack targets. Instead, DOTS merely relies on the mutual authentication and the pre-established (service) relationship between the entity owning the DOTS client and the entity owning the DOTS server. The entity owning the DOTS server SHOULD limit the attack targets that a particular DOTS client can request mitigation for as part of establishing this relationship. The method of such limitation is not in scope for this document.

Although co-location of DOTS server and mitigator within the same entity is expected to be a common deployment model, it is assumed that operators may require alternative models. Nothing in this document precludes such alternatives.

2.2. DOTS Agent Relationships

So far, we have only considered a relatively simple scenario of a single DOTS client associated with a single DOTS server, however DOTS supports more advanced relationships.

A DOTS server may be associated with one or more DOTS clients, and those DOTS clients may belong to different entities. An example scenario is a mitigation provider serving multiple attack targets (Figure 3):

```
      +---+
      | c |-----------
      +---+           \
                       \
      +---+             \ +---+
      | c |---------------| S |
      +---+             / +---+
                       /
      +---+           /
      | c |-----------
      +---+
      example.com/.org   example.net
      DOTS Clients       DOTS Server
```

Figure 3: Multiple DOTS clients for a DOTS server

A DOTS client may be associated with one or more DOTS servers, and those DOTS servers may belong to different entities. This may be to ensure high availability or co-ordinate mitigation with more than one directly connected ISP. An example scenario is for an enterprise to have DDoS mitigation service from multiple providers, as shown in Figure 4 below. Operational considerations relating to co-ordinating multiple provider responses are beyond the scope of DOTS.

[[EDITOR'S NOTE: we request working group feedback and discussion of operational considerations relating to coordinating multiple provider responses to a mitigation request.]]

```
                          +---+
              ------------| S |
             /            +---+
            /
      +---+/              +---+
      | c |---------------| S |
      +---+\              +---+
            \
             \            +---+
              ------------| S |
                          +---+
      example.com        example.net/.org
      DOTS Client        DOTS Servers
```

Figure 4: Multi-Homed DOTS Client

DOTS Relays may be either server-side or client-side, or both. A DOTS server-side relay belongs to the entity owning the DOTS server. A relay will terminate multiple discrete client connections as if it were a server and may aggregate these into a single (Figure 5) or multiple DOTS signaling sessions (Figure 6) depending upon locally applied policy. A relay will function as a server to its downstream agents and as a client to its upstream agents. Aside from the exceptions discussed in Section 4.2.2 below, the relationship between the relay and its upstream agents is opaque to the relayed clients. An example scenario is for an enterprise to have deployed multiple DOTS capable devices which are able to signal intra-domain using TCP [RFC0793] on un-congested links to a relay which may then transform these to a UDP [RFC0768] transport inter-domain where connection oriented transports may degrade; this applies to the signal channel only, as the data channel requires a connection-oriented transport. The relationship between the relay and its upstream agents is opaque to the relayed clients.

```
      +---+
      | c |\
      +---+ \              +---+
             \-----TCP-----| r |              +---+
      +---+                | e |              |   |
      | c |--------TCP-----| l |------UDP-----| S |
      +---+                | a |              |   |
             /-----TCP-----| y |              +---+
      +---+ /              +---+
      | c |/
      +---+
      example.com       example.com         example.net
      DOTS Clients      DOTS Relay          DOTS Server
```

Figure 5: Client-Side Relay with Aggregation

```
      +---+
      | c |\
      +---+ \              +---+
             \-----TCP-----| r |------UDP-----+---+
      +---+                | e |              |   |
      | c |--------TCP-----| l |------UDP-----| S |
      +---+                | a |              |   |
             /-----TCP-----| y |------UDP-----+---+
      +---+ /              +---+
      | c |/
      +---+
      example.com       example.com         example.net
      DOTS Clients      DOTS Relay          DOTS Server
```

Figure 6: Client-Side Relay without Aggregation

A variation of this scenario would be a DDoS mitigation provider deploying relays at their perimeter to consume signals across multiple transports and to consolidate these into a single transport suitable for the providers deployment, as shown in Figure 7 and Figure 8 below. The relationship between the relay and its upstream agents is opaque to the relayed clients.

[[EDITOR'S NOTE: we request working group feedback and discussion of DOTS client visibility into relayed signaling.]]

```
      +---+
      | c |\
      +---+ \             +---+
             \-----UDP-----| r |             +---+
      +---+               | e |             |   |
      | c |--------TCP-----| l |------TCP-----| S |
      +---+               | a |             |   |
             /-----TCP-----| y |             +---+
      +---+ /             +---+
      | c |/
      +---+
      example.com       example.net         example.net
      DOTS Clients      DOTS Relay          DOTS Server
```

Figure 7: Server-Side Relay with Aggregation

```
      +---+
      | c |\
      +---+ \             +---+
             \-----UDP-----| r |------TCP-----+---+
      +---+               | e |             |   |
      | c |--------TCP-----| l |------TCP-----| S |
      +---+               | a |             |   |
             /-----UDP-----| y |------TCP-----+---+
      +---+ /             +---+
      | c |/
      +---+
      example.com       example.net         example.net
      DOTS Clients      DOTS Relay          DOTS Server
```

Figure 8: Server-Side Relay without Aggregation

In the context of relays, sessions are established directly between peer DOTS agents and may not be end-to-end. In spite of this distinction a method must exist to uniquely identify the originating DOTS client. The relay should identify itself as such to any clients or servers it interacts with. Greater abstraction by way of additional layers of relays may introduce undesired complexity in regard to authentication and authorization and should be avoided.

[[EDITOR'S NOTE: we request working group feedback and discussion of the many-to-one and one-to-many client/server, client/relay, and relay/server relationships described above. We additionally request working group feedback and discussion of end-to-end signaling considerations in the context of relayed signaling.]]

3. Components

The architecture in this document is comprised of a few basic components on top of the assumed underlay network or networks described above. When connected to one another, the components represent an operational DOTS architecture.

This section describes the components themselves. Section 4 below describes the architectural concepts involved.

3.1. DOTS client

A DOTS client is a DOTS agent from which requests for help coordinating attack response originate. The requests may be in response to an active, ongoing attack against a target in the DOTS client's domain, but no active attack is required for a DOTS client to request help. Local operators may wish to have upstream traffic scrubbers in the network path for an indefinite period, and are restricted only by business relationships when it comes to duration and scope of requested mitigation.

The DOTS client requests attack response coordination from a DOTS server over the signal channel, including in the request the DOTS client's desired mitigation scoping, as described in [I-D.ietf-dots-requirements]. The actual mitigation scope and countermeasures used in response to the attack are up to the DOTS server and Mitigator operators, as the DOTS client may have a narrow perspective on the ongoing attack. As such, the DOTS client's request for mitigation should be considered advisory: guarantees of DOTS server availability or mitigation capacity constitute service level agreements and are out of scope for this document.

The DOTS client adjusts mitigation scope and provides available attack details at the direction of its local operator. Such direction may involve manual or automated adjustments in response to feedback from the DOTS server.

To provide a metric of signal health and distinguish an idle signaling session from a disconnected or defunct session, the DOTS client sends a heartbeat over the signal channel to maintain its half of the signaling session. The DOTS client similarly expects a heartbeat from the DOTS server, and MAY consider a signaling session terminated in the extended absence of a DOTS server heartbeat.

3.2. DOTS server

A DOTS server is a DOTS agent capable of receiving, processing and possibly acting on requests for help coordinating attack response from one or more DOTS clients. The DOTS server authenticates and authorizes DOTS clients as described in Signaling Sessions below, and maintains signaling session state, tracking requests for mitigation, reporting on the status of active mitigations, and terminating signaling sessions in the extended absence of a client heartbeat or when a session times out.

Assuming the preconditions discussed below exist, a DOTS client maintaining an active signaling session with a DOTS server may reasonably expect some level of mitigation in response to a request for coordinated attack response.

The DOTS server enforces authorization of DOTS clients' signals for mitigation. The mechanism of enforcement is not in scope for this document, but is expected to restrict requested mitigation scope to addresses, prefixes, and/or services owned by the DOTS client's administrative entity, such that a DOTS client from one entity is not able to influence the network path to another entity. A DOTS server MUST reject requests for mitigation of resources not owned by the requesting DOTS client's administrative entity. A DOTS server MAY also refuse a DOTS client's mitigation request for arbitrary reasons, within any limits imposed by business or service level agreements between client and server domains. If a DOTS server refuses a DOTS client's request for mitigation, the DOTS server SHOULD include the refusal reason in the server signal sent to the client.

A DOTS server is in regular contact with one or more mitigators. If a DOTS server accepts a DOTS client's request for help, the DOTS server forwards a translated form of that request to the mitigator or mitigators responsible for scrubbing attack traffic. Note that the form of the translated request passed from the DOTS server to the mitigator is not in scope: it may be as simple as an alert to mitigator operators, or highly automated using vendor or open application programming interfaces supported by the mitigator. The DOTS server MUST report the actual scope of any mitigation enabled on behalf of a client.

The DOTS server SHOULD retrieve available metrics for any mitigations activated on behalf of a DOTS client, and SHOULD include them in server signals sent to the DOTS client originating the request for mitigation.

To provide a metric of signal health and distinguish an idle signaling session from a disconnected or defunct session, the DOTS

server sends a heartbeat over the signal channel to maintain its half of the signaling session. The DOTS server similarly expects a heartbeat from the DOTS client, and MAY consider a signaling session terminated in the extended absence of a DOTS client heartbeat.

4. Concepts

4.1. Signaling Sessions

In order for DOTS to be effective as a vehicle for DDoS mitigation requests, one or more DOTS clients must establish ongoing communication with one or more DOTS servers. While the preconditions for enabling DOTS in or among network domains may also involve business relationships, service level agreements, or other formal or informal understandings between network operators, such considerations are out of scope for this document.

An established communication layer between DOTS agents is a Signaling Session. At its most basic, for a DOTS signaling session to exist both signal channel and data channel must be functioning between DOTS agents. That is, under nominal network conditions, signals actively sent from a DOTS client are received by the specific DOTS server intended by the client, and vice versa.

4.1.1. Preconditions

Prior to establishing a signaling session between agents, the owners of the networks, domains, services or applications involved are assumed to have agreed upon the terms of the relationship involved. Such agreements are out of scope for this document, but must be in place for a functional DOTS architecture.

It is assumed that as part of any DOTS service agreement, the DOTS client is provided with all data and metadata required to establish communication with the DOTS server. Such data and metadata would include any cryptographic information necessary to meet the message confidentiality, integrity and authenticity requirement in [I-D.ietf-dots-requirements], and might also include the pool of DOTS server addresses and ports the DOTS client should use for signal and data channel messaging.

4.1.2. Establishing the Signaling Session

With the required business or service agreements in place, the DOTS client initiates a signal session by contacting the DOTS server over the signal channel and the data channel. To allow for DOTS service flexibility, neither the order of contact nor the time interval

between channel creations is specified. A DOTS client MAY establish signal channel first, and then data channel, or vice versa.

The methods by which a DOTS client receives the address and associated service details of the DOTS server are not prescribed by this document. For example, a DOTS client may be directly configured to use a specific DOTS server address and port, and directly provided with any data necessary to satisfy the Peer Mutual Authentication requirement in [I-D.ietf-dots-requirements], such as symmetric or asymmetric keys, usernames and passwords, etc. All configuration and authentication information in this scenario is provided out-of-band by the entity operating the DOTS server.

At the other extreme, the architecture in this document allows for a form of DOTS client auto-provisioning. For example, the entity operating the DOTS server or servers might provide the client entity only with symmetric or asymmetric keys to authenticate the provisioned DOTS clients. Only the keys would then be directly configured on DOTS clients, but the remaining configuration required to provision the DOTS clients could be learned through mechanisms similar to DNS SRV [RFC2782] or DNS Service Discovery [RFC6763].

The DOTS client SHOULD successfully authenticate and exchange messages with the DOTS server over both signal and data channel as soon as possible to confirm that both channels are operational.

Once the DOTS client begins receiving DOTS server signals, the signaling session is active. At any time during the signaling session, the DOTS client MAY use the data channel to adjust initial configuration, manage black- and white-listed prefixes or addresses, leverage vendor-specific extensions, and so on. Note that unlike the signal channel, there is no requirement that the data channel remain operational in attack conditions (See Data Channel Requirements, [I-D.ietf-dots-requirements]).

4.1.3. Maintaining the Signaling Session

DOTS clients, servers and relays periodically send heartbeats to each other over the signal channel, per Operational Requirements discussed in [I-D.ietf-dots-requirements]. DOTS agent operators SHOULD configure the heartbeat interval such that the frequency does not lead to accidental denials of service due to the overwhelming number of heartbeats a DOTS agent must field.

Either DOTS agent may consider a signaling session terminated in the extended absence of a heartbeat from its peer agent. The period of that absence will be established in the protocol definition.

4.2. Modes of Signaling

This section examines the modes of signaling between agents in a DOTS architecture.

4.2.1. Direct Signaling

A signaling session may take the form of direct signaling between the DOTS clients and servers, as shown in Figure 9 below:

```
        +-------------+                            +-------------+
        | DOTS client |<------signal session------>| DOTS server |
        +-------------+                            +-------------+
```

Figure 9: Direct Signaling

In a direct signaling session, DOTS client and server are communicating directly, with no relays in the signaling path. A direct signaling session MAY exist inter- or intra-domain. The signaling session is abstracted from the underlying networks or network elements the signals traverse: in a direct signaling session, the DOTS client and server are logically peer DOTS agents.

4.2.2. Relayed Signaling

A signaling session may also include one or more DOTS relays in the signaling path between the clients and servers, as shown in Figure 10:

```
    +-------------+                             +-------------+
    | DOTS client |                             | DOTS server |
    +-------------+                             +-------------+
           ^                                           ^
           |    +------------+      +------------+    |
           +--->| DOTS relay |<----->| DOTS relay |<---+
                +------------+      +------------+
```

Figure 10: Relayed Signaling

To allow for maximum architectural flexibility, no restriction is placed on the number of relays in the signaling path. Operators of DOTS agents should consider the impact on signal latency incurred by each additional DOTS relay in the signaling path, as well as the increased operational complexity, when deploying DOTS relays.

[[EDITOR'S NOTE: we request working group feedback and discussion of operational considerations related to DOTS relays, particularly with respect to the implications of multiple relays in the signal path.]]

As discussed above in Section 2.2, relays may be client-side or server-side. In either case, the relay appears to the peer agent as its logical opposite. That is, a DOTS relay appears to a DOTS client or downstream relay as a DOTS server. Conversely, a DOTS relay appears to a DOTS server or upstream DOTS relay as a DOTS client. Thus relayed signaling may be thought of as chained direct signaling sessions.

4.2.3. Redirected Signaling

In certain circumstances, a DOTS server may want to redirect a DOTS client to an alternative DOTS server for a signaling session. Such circumstances include but are not limited to:

- o Maximum number of signaling sessions with clients has been reached;
- o Mitigation capacity exhaustion in the Mitigator with which the specific DOTS server is communicating;
- o Mitigator outage or other downtime, such as scheduled maintenance;
- o Scheduled DOTS server maintenance;
- o Scheduled modifications to the network path between DOTS server and DOTS client.

A basic redirected signaling session resembles the following, as shown in Figure 11:

```
        +-------------+                            +---------------+
        |             |<-(1)-- signal session 1 -->|               |
        |             |                            |               |
        |             |<=(2)== redirect to B ======|               |
        | DOTS client |                            | DOTS server A |
        |             |X-(4)-- signal session 1 --X|               |
        |             |                            |               |
        |             |                            |               |
        +-------------+                            +---------------+
               ^
               |
              (3) signal session 2
               |
               v
        +---------------+
        | DOTS server B |
        +---------------+
```

Figure 11: Redirected Signaling

1. Previously established signaling session 1 exists between a DOTS client and DOTS server with address A.

2. DOTS server A sends a server signal redirecting the client to DOTS server B.

3. If the DOTS client does not already have a separate signaling session with the redirection target, the DOTS client initiates and establishes a signaling session with DOTS server B as described above.

4. Having redirected the DOTS client, DOTS server A ceases sending server signals. The DOTS client likewise stops sending client signals to DOTS server A. Signal session 1 is terminated.

[[EDITOR'S NOTE: we request working group feedback and discussion of the need for redirected signaling.]]

4.2.4. Recursive Signaling

DOTS is centered around improving the speed and efficiency of coordinated response to DDoS attacks. One scenario not yet discussed involves coordination among federated entities operating DOTS servers and mitigators.

In the course of normal DOTS operations, a DOTS client communicates the need for mitigation to a DOTS server, and that server initiates mitigation on a mitigator with which the server has an established

service relationship. The operator of the mitigator may in turn monitor mitigation performance and capacity, as the attack being mitigated may grow in severity beyond the mitigating entity's capabilities.

The operator of the mitigator has limited options in the event a DOTS client-requested mitigation is being overwhelmed by the severity of the attack. Out-of-scope business or service level agreements may permit the mitigating entity to drop the mitigation and let attack traffic flow unchecked to the target, but this is only encourages attack escalation. In the case where the mitigating entity is the upstream service provider for the attack target, this may mean the mitigating entity and its other services and users continue to suffer the incidental effects of the attack.

A recursive signaling model as shown in Figure 12 below offers an alternative. In a variation of the primary use case "Successful Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services" described in [I-D.ietf-dots-use-cases], an entity operating a DOTS server and mitigation has a mitigator that is itself capable of acting as a DOTS client. The mitigator with DOTS client capabilities has an established signaling session with a DOTS server belonging to a separate administrative entity.

With these preconditions in place, the operator of the mitigator being overwhelmed or otherwise performing inadequately may request mitigation for the attack target from this separate DOTS-aware entity. Such a request recurses the originating mitigation request to the secondary DOTS server, in the hope of building a cumulative mitigation against the attack:

```
                     example.net entity
                     . . . . . . . . . . . . . . . . .
                     .                               .
       +----+    A   .  +----+       +-----------+   .
       | Cc |<--------->| Sn |~~~~~~~| Mitigator |   .
       +----+        .  +----+       |     Mn    |   .
                     .               |   +----+  |   .
     example.com     .               +---| Cn |--+   .
        client       .                   +----+      .
                     .                     ^         .
                     . . . . . . . . . . . | . . . . .
                                           |
                                           | B
                                           |
                     . . . . . . . . . . . | . . . . .
                     .                     v         .
                     .  +-----------+    +----+      .
                     .  | Mitigator |~~~~| So |      .
                     .  |     Mo    |    +----+      .
                     .  +-----------+                .
                     .                               .
                     . . . . . . . . . . . . . . . . .
                     example.org entity
```

Figure 12: Recursive Signaling

In Figure 12 above, client Cc signals a request for mitigation across inter-domain signaling session A to the DOTS server Sn belonging to the example.net entity. DOTS server Sn enables mitigation on mitigator Mn, which, acting as DOTS client Cn, has pre-existing inter-domain signaling session B with the DOTS server So belonging to the example.org entity. At any point, DOTS client Cn MAY recurse an on-going mitigation request to DOTS server So, in the expectation that mitigator Mo will be activated to aid in the defense of the attack target.

Recursive signaling is opaque to the DOTS client. To maximize mitigation visibility to the DOTS client, however, the recursing entity SHOULD provide recursed mitigation feedback in signals reporting on mitigation status to the DOTS client. For example, the recursing entity's mitigator should incorporate into mitigation status messages available metrics such as dropped packet or byte counts from the recursed mitigation.

DOTS clients involved in recursive signaling MUST be able to withdraw requests for mitigation without warning or justification, per [I-D.ietf-dots-requirements].

Operators of recursing mitigators MAY maintain the recursed mitigation for a brief, protocol-defined period in the event the DOTS client originating the mitigation withdraws its request for help, as per the discussion of managing mitigation toggling in the operational requirements ([I-D.ietf-dots-requirements]). Service or business agreements between recursing entities are not in scope for this document.

[[EDITOR'S NOTE: Recursive signaling raises questions about how to authenticate and authorize the recursed request, how end-to-end signaling functions in such a scenario, and implications for operational and data privacy, as well as what level of visibility a client has into the recursed mitigation. We ask the working group for feedback and additional discussion of these issues to help settle the way forward.]]

5. Security Considerations

This section describes identified security considerations for the DOTS architecture.

DOTS is at risk from three primary attack vectors: agent impersonation, traffic injection and signal blocking. These vectors may be exploited individually or in concert by an attacker to confuse, disable, take information from, or otherwise inhibit the DOTS system.

Any attacker with the ability to impersonate a legitimate client or server or, indeed, inject false messages into the stream may potentially trigger/withdraw traffic redirection, trigger/cancel mitigation activities or subvert black/whitelists. From an architectural standpoint, operators SHOULD ensure best current practices for secure communication are observed for data and signal channel confidentiality, integrity and authenticity. Care must be taken to ensure transmission is protected by appropriately secure means, reducing attack surface by exposing only the minimal required services or interfaces. Similarly, received data at rest SHOULD be stored with a satisfactory degree of security.

As many mitigation systems employ diversion to scrub attack traffic, operators of DOTS agents SHOULD ensure signaling sessions are resistant to Man-in-the-Middle (MitM) attacks. An attacker with control of a DOTS client or relay may negatively influence network traffic by requesting and withdrawing requests for mitigation for particular prefixes, leading to route or DNS flapping.

Any attack targeting the availability of DOTS servers may disrupt the ability of the system to receive and process DOTS signals resulting

in failure to fulfill a mitigation request. Similarly, DOTS relays represent high-value targets in a DOTS architecture. Disrupting any DOTS relay in a signaling path represents a denial-of-service against DOTS in general. DOTS systems SHOULD be given adequate protections, again, in accordance with best current practices for network and host security.

6. Acknowledgments

Thanks to Matt Richardson for last minute comments and suggestions.

7. Change Log

2016-03-18 Initial revision

8. References

8.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[I-D.ietf-dots-requirements]
Mortensen, A., Moskowitz, R., and T. Reddy, "DDoS Open Threat Signaling Requirements", draft-ietf-dots-requirements-00 (work in progress), October 2015.

[I-D.ietf-dots-use-cases]
Dobbins, R., Fouant, S., Migault, D., Moskowitz, R., Teague, N., and L. Xia, "Use cases for DDoS Open Threat Signaling", draft-ietf-dots-use-cases-00 (work in progress), October 2015.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <http://www.rfc-editor.org/info/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <http://www.rfc-editor.org/info/rfc2782>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

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

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

Authors' Addresses

Andrew Mortensen
Arbor Networks, Inc.
2727 S. State St
Ann Arbor, MI 48104
United States

EMail: amortensen@arbor.net

Flemming Andreasen
Cisco Systems, Inc.
United States

EMail: fandreas@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

EMail: tireddy@cisco.com

Christopher Gray
Comcast, Inc.
United States

EMail: Christopher_Gray3@cable.comcast.com

Rich Compton
Charter Communications, Inc.

EMail: Rich.Compton@charter.com

Nik Teague
Verisign, Inc.
United States

EMail: nteague@verisign.com