Internet Engineering Task Force (IETF)
Request for Comments: 8811
Category: Informational
ISSN: 2070-1721

A. Mortensen, Ed.
Forcepoint
T. Reddy.K, Ed.
McAfee, Inc.
F. Andreasen
Cisco
N. Teague
Iron Mountain
R. Compton
Charter

August 2020

DDoS Open Threat Signaling (DOTS) Architecture

Abstract

This document describes an architecture for establishing and maintaining Distributed Denial-of-Service (DDoS) Open Threat Signaling (DOTS) within and between domains. The document does not specify protocols or protocol extensions, instead focusing on defining architectural relationships, components, and concepts used in a DOTS deployment.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8811.

Table of Contents

1. Context and Motivation

Signaling the need for help to defend against an active distributed denial-of-service (DDoS) attack requires a common understanding of mechanisms and roles among the parties coordinating a defensive response. The signaling layer and supplementary messaging are the focus of DDoS Open Threat Signaling (DOTS). DOTS defines a method of coordinating defensive measures among willing peers to mitigate attacks quickly and efficiently, enabling hybrid attack responses coordinated locally at or near the target of an active attack, or anywhere in path between attack sources and target. Sample DOTS use cases are elaborated in [DOTS-USE-CASES].

This document describes an architecture used in establishing, maintaining, or terminating a DOTS relationship within a domain or between domains.

1.1. Terminology

1.1.1. Key Words

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.

1.1.2. Definition of Terms

This document uses the terms defined in [RFC8612].

1.2. Scope

In this architecture, DOTS clients and servers communicate using DOTS signal channel [RFC8782] and data channel [RFC8783] protocols.

The DOTS architecture presented here is applicable across network administrative domains, for example, between an enterprise domain and the domain of a third-party attack mitigation 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 networks, but single domain scenarios are valuable in their own right, as when aggregating intra-domain DOTS client signals for an inter-domain coordinated attack response.

This document does not address any administrative or business agreements that may be established between involved DOTS parties. Those considerations are out of scope. Regardless, this document assumes necessary authentication and authorization mechanisms are put in place so that only authorized clients can invoke the DOTS service.

A detailed set of DOTS requirements are discussed in [RFC8612], and the DOTS architecture is designed to follow those requirements. Only new behavioral requirements are described in this document.

1.3. Assumptions

This document makes the following assumptions:

* All domains 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, in general, conditions should be assumed to be hostile, and DOTS must be able to function in all circumstances, including when the signaling path is significantly impaired. Congestion control requirements are discussed in Section 3 of [RFC8612]. The DOTS signal channel defined in [RFC8782] is designed to be extremely resilient under extremely hostile network conditions, and it provides continued contact between DOTS agents even as DDoS attack traffic saturates the link.

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

* Mitigation requests may be sent to one or more upstream DOTS servers based on criteria determined by DOTS client administrators and the underlying network configuration. The number of DOTS servers with which a given DOTS client has established communications is determined by local policy and is deployment specific. For example, a DOTS client of a multihomed network may support built-in policies to establish DOTS relationships with DOTS servers located upstream of each interconnection link.

* The mitigation capacity and/or capability of domains receiving requests for coordinated attack response is opaque to the domains sending the request. The domain 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. In either case, the upstream DOTS server may redirect a request to another DOTS server. Redirection may be local to the redirecting DOTS

server's domain or may involve a third-party domain.

* 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 domain and the DOTS server domain. Any encapsulation required for successful delivery is left untouched by transit network elements. DOTS servers and DOTS clients cannot assume any preferential treatment of DOTS signals. Such preferential treatment may be available in some deployments (e.g., intra-domain scenarios), and the DOTS architecture does not preclude its use when available. However, DOTS itself does not address how that may be done.

* 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 the DOTS architecture.

* The signal and data channels are loosely coupled and might not terminate on the same DOTS server. How the DOTS servers synchronize the DOTS configuration is out of scope of this specification.

2. DOTS Architecture

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

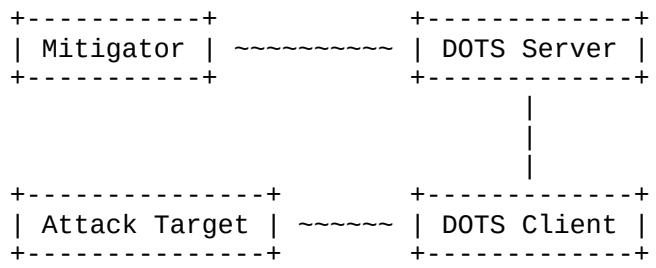


Figure 1: Basic DOTS Architecture

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 service provided by the mitigator is called "DDoS mitigation service". 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. In turn, the DOTS server 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 and DOTS server. 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 is out of scope of DOTS. Thus, DOTS specifies neither how an attack target decides it is under DDoS attack nor does DOTS specify how a mitigator may actually mitigate such an attack. A DOTS client's request for mitigation is advisory in nature and might not lead to any mitigation at all, depending on the DOTS server domain's capacity and willingness to mitigate on behalf of the DOTS client domain.

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 sessions by connecting to the provided DOTS server addresses.

As illustrated in Figure 2, there are two interfaces between a DOTS server and a DOTS client: a signal channel and (optionally) a data channel.

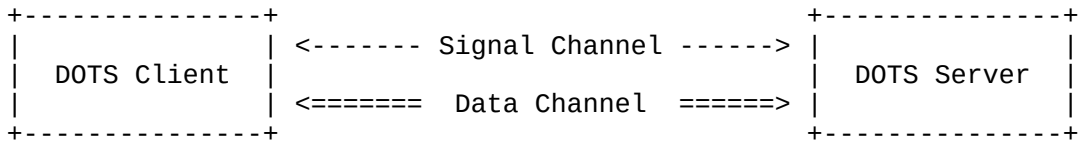


Figure 2: DOTS Interfaces

The primary purpose of the signal channel is for a DOTS client to ask a 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 that contains information about the attack target(s). The client signal may also include telemetry information about the attack, if the DOTS client has such information available. In turn, the DOTS server 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 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; it should also include efficacy information about the attack mitigation 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 lifetime is exceeded. 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, more efficient capabilities. 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, but are not limited to:

* Creating identifiers, such as names or aliases, for resources for which mitigation may be requested. Such identifiers may then be used in subsequent signal channel exchanges to refer more efficiently to the resources under attack.

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

* Accept-list management, which enables a DOTS client to inform the DOTS server about sources from which traffic is always accepted.

* Filter management, which enables a DOTS client to install or remove traffic filters dropping or rate-limiting unwanted traffic.

* DOTS client provisioning.

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

this means that DOTS deployments should rely on such information being exchanged only under normal traffic conditions.

2.1. DOTS Operations

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 a domain that has an identity that can be authenticated and authorized. DOTS agents communicate with each other over a mutually authenticated signal channel and (optionally) 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.

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 bidirectional communication is possible at this time between the DOTS client and DOTS server. Furthermore, it is 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 active. 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 with the help of an upstream mitigator may involve some form of traffic redirection whereby traffic destined for the attack target is steered towards the mitigator. Common mechanisms to achieve this redirection depend on BGP [RFC4271] and DNS [RFC1035]. In turn, the mitigator inspects and scrubs the traffic and forwards the resulting (hopefully non-attack) traffic to the attack target. 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.

DOTS relies on mutual authentication and the pre-established service relationship between the DOTS client domain and the DOTS server domain to provide authorization. The DOTS server should enforce authorization mechanisms to restrict the mitigation scope a DOTS client can request, but such authorization mechanisms are deployment specific.

Although co-location of DOTS server and mitigator within the same domain 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. Components

2.2.1. DOTS Client

A DOTS client is a DOTS agent from which requests for help coordinating an attack response originate. The requests may be in response to an active, ongoing attack against a target in the DOTS client domain, but no active attack is required for a DOTS client to request help. Operators may wish to have upstream mitigators 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 [RFC8612] (SIG-008). 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 (SLAs) and are out of scope for this document.

The DOTS client adjusts mitigation scope and provides available mitigation feedback (e.g., mitigation efficacy) at the direction of its local administrator. Such direction may involve manual or automated adjustments in response to updates from the DOTS server.

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

2.2.2. DOTS Server

A DOTS server is a DOTS agent capable of receiving, processing, and possibly acting on requests for help coordinating attack responses from DOTS clients. The DOTS server authenticates and authorizes DOTS clients as described in Section 3.1 and maintains session state, tracks requests for mitigation, reports on the status of active mitigations, and terminates 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 session with a DOTS server may reasonably expect some level of mitigation in response to a request for coordinated attack response.

For a given DOTS client (administrative) domain, the DOTS server needs to be able to determine whether a given resource is in that domain. For example, this could take the form of associating a set of IP addresses and/or prefixes per DOTS client domain. The DOTS server enforces authorization of signals for mitigation, filtering rules, and aliases for resources from DOTS clients. The mechanism of enforcement is not in scope for this document but is expected to restrict mitigation requests, filtering rules, aliases for addresses and prefixes, and/or services owned by the DOTS client domain, such that a DOTS client from one domain is not able to influence the network path to another domain. A DOTS server MUST reject mitigation requests, filtering rules, and aliases for resources not owned by the requesting DOTS client's administrative domain. The exact mechanism for the DOTS servers to validate that the resources are within the scope of the DOTS client domain is deployment specific. For example, if the DOTS client domain uses Provider-Aggregatable prefixes for its resources and leverages the DDoS mitigation service of the Internet Transit Provider (ITP); the ITP knows the prefixes assigned to the

DOTS client domain because they are assigned by the ITP itself. However, if the DDoS Mitigation is offered by a third-party DDoS mitigation service provider; it does not know the resources owned by the DOTS client domain. The DDoS mitigation service provider and the DOTS client domain can opt to use the identifier validation challenges discussed in [RFC8555] and [RFC8738] to identify whether or not the DOTS client domain actually controls the resources. The challenges for validating control of resources must be performed when no attack traffic is present and works only for "dns" and "ip" identifier types. Further, if the DOTS client lies about the resources owned by the DOTS client domain, the DDoS mitigation service provider can impose penalties for violating the SLA. A DOTS server MAY also refuse a DOTS client's mitigation request for arbitrary reasons, within any limits imposed by business or SLAs between client and server domains. If a DOTS server refuses a DOTS client's request for mitigation, the DOTS server MUST 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(s) 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 signal channel from a disconnected or defunct channel, the DOTS server MUST send a heartbeat over the signal channel to maintain its half of the channel. The DOTS server similarly expects a heartbeat from the DOTS client and MAY consider a session terminated in the extended absence of a DOTS client heartbeat.

2.2.3. DOTS Gateway

Traditional client/server relationships may be expanded by chaining DOTS sessions. This chaining is enabled through "logical concatenation" of a DOTS server and a DOTS client, resulting in an application analogous to the Session Initiation Protocol (SIP) [RFC3261] logical entity of a Back-to-Back User Agent (B2BUA) [RFC7092]. The term "DOTS gateway" is used here in the descriptions of selected scenarios involving this application.

A DOTS gateway may be deployed client side, server side, or both. The gateway may terminate multiple discrete client connections and may aggregate these into a single or multiple DOTS session(s).

The DOTS gateway will appear as a server to its downstream agents and as a client to its upstream agents, a functional concatenation of the DOTS client and server roles, as depicted in Figure 3:

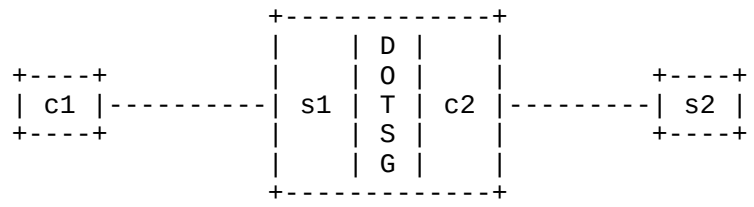


Figure 3: DOTS Gateway

The DOTS gateway MUST perform full stack DOTS session termination and reorigination between its client and server side. The details of how this is achieved are implementation specific.

2.3. 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 domains. An example scenario is a mitigation provider serving multiple attack targets (Figure 4).

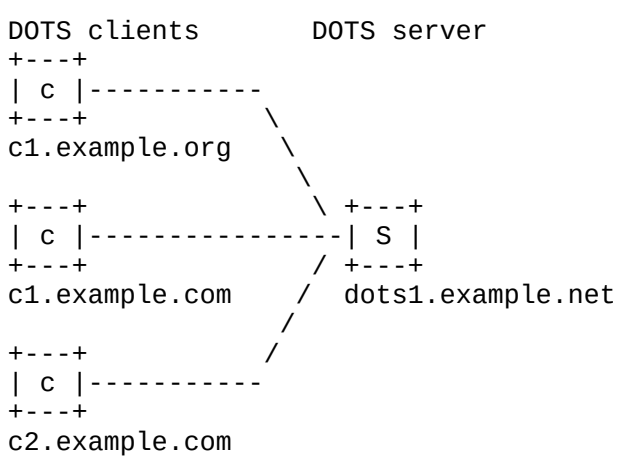


Figure 4: DOTS Server with Multiple Clients

A DOTS client may be associated with one or more DOTS servers, and those DOTS servers may belong to different domains. This may be to ensure high availability or coordinate 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 5.

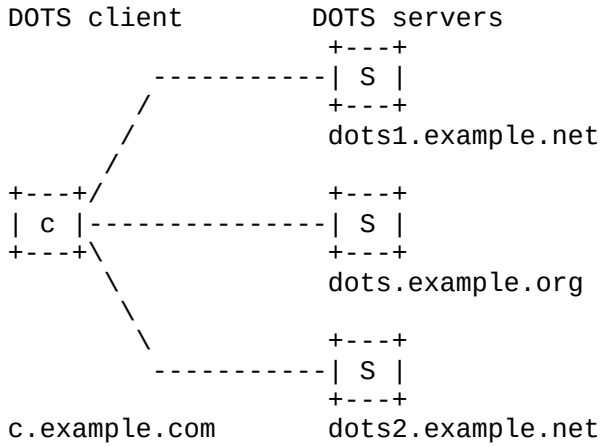


Figure 5: Multihomed DOTS Client

Deploying a multihomed client requires extra care and planning, as the DOTS servers with which the multihomed client communicates might not be affiliated. Should the multihomed client simultaneously request for mitigation from all servers with which it has established signal channels, the client may unintentionally inflict additional network disruption on the resources it intends to protect. In one of

the worst cases, a multihomed DOTS client could cause a permanent routing loop of traffic destined for the client's protected services, as the uncoordinated DOTS servers' mitigators all try to divert that traffic to their own scrubbing centers.

The DOTS protocol itself provides no fool-proof method to prevent such self-inflicted harms as a result of deploying multihomed DOTS clients. If DOTS client implementations nevertheless include support for multihoming, they are expected to be aware of the risks, and consequently to include measures aimed at reducing the likelihood of negative outcomes. Simple measures might include:

* Requesting mitigation serially, ensuring only one mitigation request for a given address space is active at any given time;

* Dividing the protected resources among the DOTS servers, such that no two mitigators will be attempting to divert and scrub the same traffic;

* Restricting multihoming to deployments in which all DOTS servers are coordinating management of a shared pool of mitigation resources.

2.3.1. Gatewayed Signaling

As discussed in Section 2.2.3, a DOTS gateway is a logical function chaining DOTS sessions through concatenation of a DOTS server and DOTS client.

An example scenario, as shown in Figure 6 and Figure 7, is for an enterprise to have deployed multiple DOTS-capable devices that are able to signal intra-domain using TCP [RFC0793] on uncongested links to a DOTS gateway that 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 gateway and its upstream agents is opaque to the initial clients.

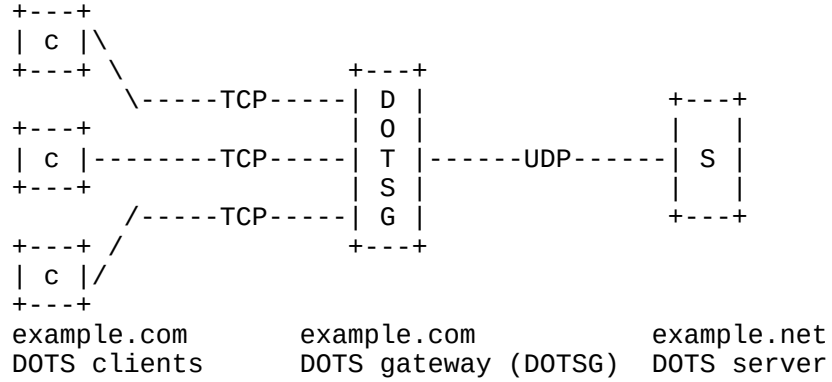
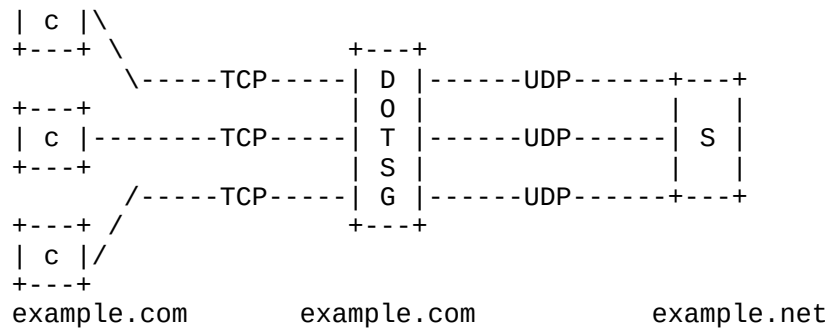


Figure 6: Client-Side Gateway with Aggregation



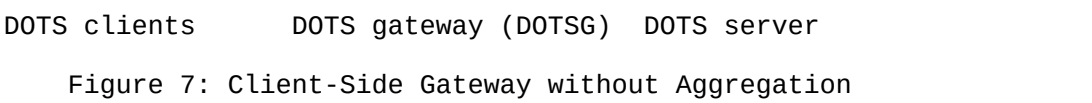


Figure 7: Client-Side Gateway without Aggregation

This may similarly be deployed in the inverse scenario where the gateway resides in the server-side domain and may be used to terminate and/or aggregate multiple clients to a single transport as shown in Figure 8 and Figure 9.

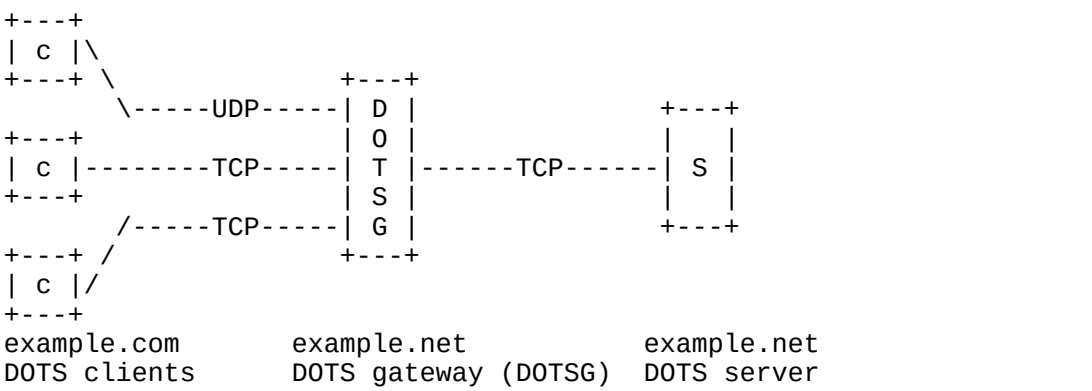


Figure 8: Server-Side Gateway with Aggregation

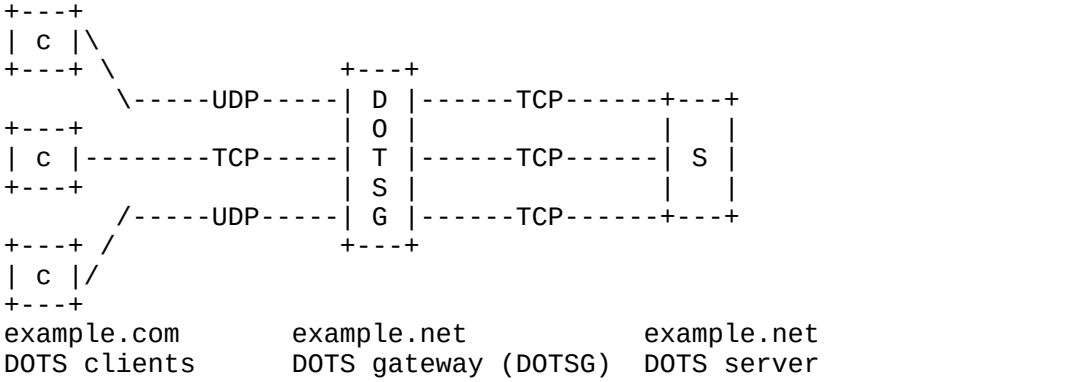


Figure 9: Server-Side Gateway without Aggregation

This document anticipates scenarios involving multiple DOTS gateways. An example is a DOTS gateway at the network client's side and another one at the server side. The first gateway can be located at Customer Premises Equipment (CPE) to aggregate requests from multiple DOTS clients enabled in an enterprise network. The second DOTS gateway is deployed on the provider side. This scenario can be seen as a combination of the client-side and server-side scenarios.

3. Concepts

3.1. DOTS 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, SLAs, or other formal or informal understandings between network operators, such considerations are out of scope for this document.

A DOTS session is established to support bilateral exchange of data between an associated DOTS client and a DOTS server. In the DOTS architecture, data is exchanged between DOTS agents over signal and data channels. As such, a DOTS session can be a DOTS signal channel session, a DOTS data channel session, or both. The DOTS server

couples the DOTS signal and data channel sessions using the DOTS client identity. The DOTS session is further elaborated in the DOTS signal channel protocol defined in [RFC8782] and the DOTS data channel protocol defined in [RFC8783].

A DOTS agent can maintain one or more DOTS sessions.

A DOTS signal channel session is associated with a single transport connection (TCP or UDP session) and a security association (a TLS or DTLS session). Similarly, a DOTS data channel session is associated with a single TCP connection and a TLS security association.

Mitigation requests created using the DOTS signal channel are not bound to the DOTS signal channel session. Instead, mitigation requests are associated with a DOTS client and can be managed using different DOTS signal channel sessions.

3.1.1. Preconditions

Prior to establishing a DOTS 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 (SEC-002) in [RFC8612] and might also include the pool of DOTS server addresses and ports the DOTS client should use for signal and data channel messaging.

3.1.2. Establishing the DOTS Session

With the required business agreements in place, the DOTS client initiates a DOTS session by contacting its DOTS server(s) over the signal channel and (possibly) 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 the signal channel first, and then the 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 IP address and port, and be directly provided with any data necessary to satisfy the Peer Mutual Authentication requirement (SEC-001) in [RFC8612], such as symmetric or asymmetric keys, usernames, passwords, etc. All configuration and authentication information in this scenario is provided out of band by the domain 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 domain operating the DOTS server or servers might provide the client domain 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 the signal and (if used) data channel as soon as possible to confirm that both channels are

operational.

As described in [RFC8612] (DM-008), the DOTS client can configure preferred values for acceptable signal loss, mitigation lifetime, and heartbeat intervals when establishing the DOTS signal channel session. A DOTS signal channel session is not active until DOTS agents have agreed on the values for these DOTS session parameters, a process defined by the protocol.

Once the DOTS client begins receiving DOTS server signals, the DOTS session is active. At any time during the DOTS session, the DOTS client may use the data channel to manage aliases, manage drop- and accept-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 remains operational in attack conditions. (See "Data Channel Requirements" Section 2.3 of [RFC8612]).

3.1.3. Maintaining the DOTS Session

DOTS clients and servers periodically send heartbeats to each other over the signal channel, discussed in [RFC8612] (SIG-004). 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 DOTS signal channel 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.

3.2. Modes of Signaling

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

3.2.1. Direct Signaling

A DOTS session may take the form of direct signaling between the DOTS clients and servers, as shown in Figure 10.

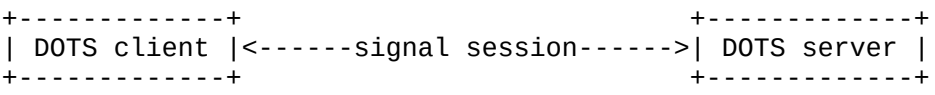


Figure 10: Direct Signaling

In a direct DOTS session, the DOTS client and server are communicating directly. Direct signaling may exist inter- or intra-domain. The DOTS session is abstracted from the underlying networks or network elements the signals traverse; in direct signaling, the DOTS client and server are logically adjacent.

3.2.2. Redirected Signaling

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

* Maximum number of DOTS signal channel sessions with clients has been reached;

* Mitigation capacity exhaustion in the mitigator with which the specific DOTS server is communicating;

* Mitigator outage or other downtime such as scheduled maintenance;

* Scheduled DOTS server maintenance;

* Scheduled modifications to the network path between DOTS server and DOTS client.

A basic redirected DOTS signal channel session resembles the following, as shown in Figure 11.

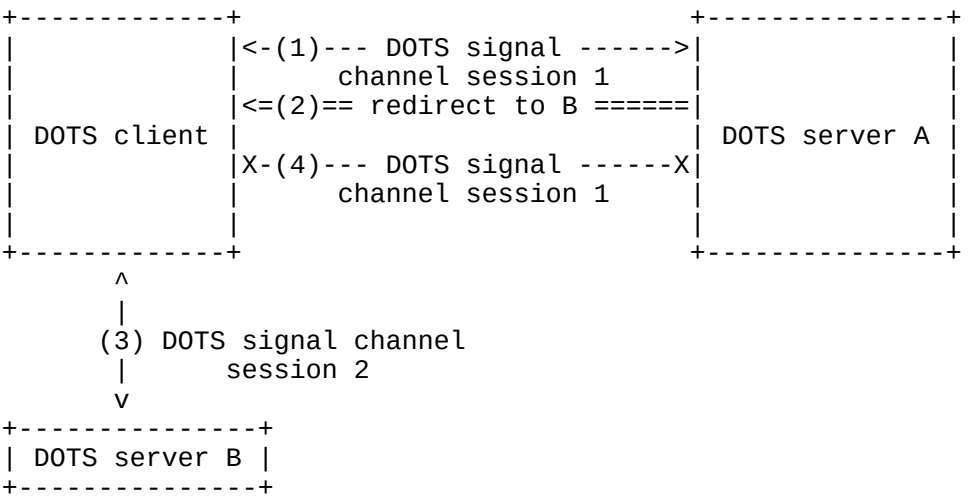


Figure 11: Redirected Signaling

1. Previously established DOTS signal channel session 1 exists between a DOTS client and DOTS server 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 DOTS signal channel session with the redirection target, the DOTS client initiates and establishes DOTS signal channel session 2 with DOTS server B.

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. DOTS signal channel session 1 is terminated.

3.2.3. Recursive Signaling

DOTS is centered around improving the speed and efficiency of a coordinated response to DDoS attacks. One scenario not yet discussed involves coordination among federated domains 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 domain’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 SLAs may permit the mitigating domain to drop the mitigation and let attack traffic flow unchecked to the target, but this only encourages attack escalation. In the case where the mitigating domain is the upstream service provider for

the attack target, this may mean the mitigating domain and its other services and users continue to suffer the incidental effects of the attack.

A recursive signaling model as shown in Figure 12 offers an alternative. In a variation of the use case "Upstream DDoS Mitigation by an Upstream Internet Transit Provider" described in [DOTS-USE-CASES], a domain operating a DOTS server and mitigator also operates a DOTS client. This DOTS client has an established DOTS session with a DOTS server belonging to a separate administrative domain.

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 domain. Such a request recurses the originating mitigation request to the secondary DOTS server in the hope of building a cumulative mitigation against the attack.

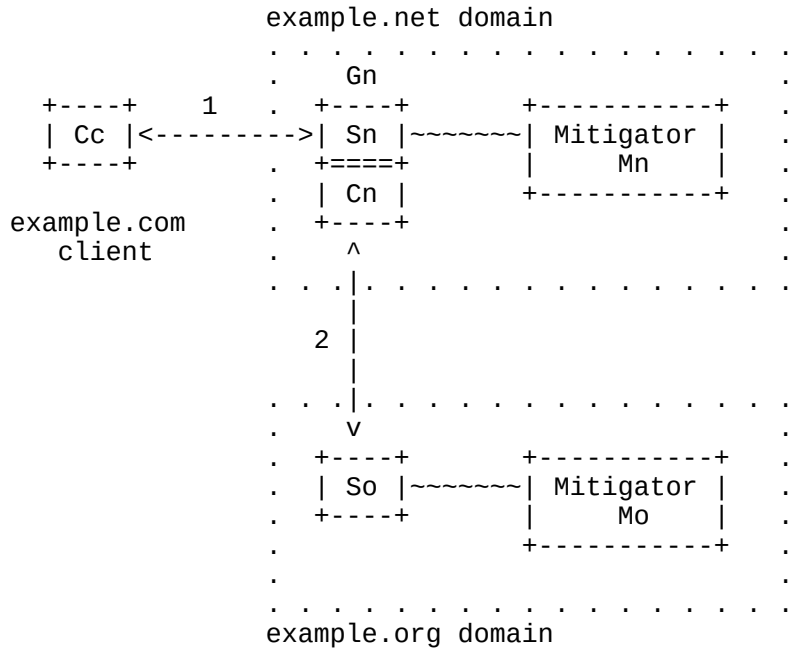


Figure 12: Recursive Signaling

In Figure 12, client Cc signals a request for mitigation across inter-domain DOTS session 1 to the DOTS server Sn belonging to the example.net domain. DOTS server Sn enables mitigation on mitigator Mn. DOTS server Sn is half of DOTS gateway Gn, being deployed logically back to back with DOTS client Cn, which has preexisting inter-domain DOTS session 2 with the DOTS server So belonging to the example.org domain. At any point, DOTS server Sn MAY recurse an ongoing mitigation request through DOTS client Cn 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 domain SHOULD provide recursed mitigation feedback in signals reporting on mitigation status to the DOTS client. For example, the recursing domain’s DOTS server should incorporate available metrics such as dropped packet or byte counts from the recursed domain’s DOTS server into mitigation status messages.

DOTS clients involved in recursive signaling must be able to withdraw requests for mitigation without warning or justification per SIG-006 in [RFC8612].

Operators recursing mitigation requests 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 SIG-006 of [RFC8612].

Deployment of recursive signaling may result in traffic redirection, examination, and mitigation extending beyond the initial bilateral relationship between DOTS client and DOTS server. As such, client control over the network path of mitigated traffic may be reduced. DOTS client operators should be aware of any privacy concerns and work with DOTS server operators employing recursive signaling to ensure shared sensitive material is suitably protected. Typically, there is a contractual SLA negotiated among the DOTS client domain, the recursed domain, and the recursing domain to meet the privacy requirements of the DOTS client domain and authorization for the recursing domain to request mitigation for the resources controlled by the DOTS client domain.

3.2.4. Anycast Signaling

The DOTS architecture does not assume the availability of anycast within a DOTS deployment, but neither does the architecture exclude it. Domains operating DOTS servers MAY deploy DOTS servers with an anycast Service Address as described in BCP 126 [RFC4786]. In such a deployment, DOTS clients connecting to the DOTS Service Address may be communicating with distinct DOTS servers, depending on the network configuration at the time the DOTS clients connect. Among other benefits, anycast signaling potentially offers the following:

* Simplified DOTS client configuration, including service discovery through the methods described in [RFC7094]. In this scenario, the "instance discovery" message would be a DOTS client initiating a DOTS session to the DOTS server anycast Service Address, to which the DOTS server would reply with a redirection to the DOTS server unicast address the client should use for DOTS.

* Region- or customer-specific deployments, in which the DOTS Service Addresses route to distinct DOTS servers depending on the client region or the customer network in which a DOTS client resides.

* Operational resiliency, spreading DOTS signaling traffic across the DOTS server domain’s networks, and thereby also reducing the potential attack surface, as described in BCP 126 [RFC4786].

3.2.4.1. Anycast Signaling Considerations

As long as network configuration remains stable, anycast DOTS signaling is to the individual DOTS client indistinct from direct signaling. However, the operational challenges inherent in anycast signaling are anything but negligible, and DOTS server operators must carefully weigh the risks against the benefits before deploying.

While the DOTS signal channel primarily operates over UDP per SIG-001 in [RFC8612], the signal channel also requires mutual authentication between DOTS agents, with associated security state on both ends.

Network instability is of particular concern with anycast signaling, as DOTS signal channels are expected to be long lived and potentially operating under congested network conditions caused by a volumetric DDoS attack.

For example, a network configuration altering the route to the DOTS server during active anycast signaling may cause the DOTS client to

send messages to a DOTS server other than the one with which it initially established a signaling session. That second DOTS server might not have the security state of the existing session, forcing the DOTS client to initialize a new DOTS session. This challenge might in part be mitigated by use of resumption via a pre-shared key (PSK) in TLS 1.3 [RFC8446] and DTLS 1.3 [DTLS-PROTOCOL] (session resumption in TLS 1.2 [RFC5246] and DTLS 1.2 [RFC6347]), but keying material must then be available to all DOTS servers sharing the anycast Service Address, which has operational challenges of its own.

While the DOTS client will try to establish a new DOTS session with the DOTS server now acting as the anycast DOTS Service Address, the link between DOTS client and server may be congested with attack traffic, making signal session establishment difficult. In such a scenario, anycast Service Address instability becomes a sort of signal session flapping, with obvious negative consequences for the DOTS deployment.

Anycast signaling deployments similarly must also take into account active mitigations. Active mitigations initiated through a DOTS session may involve diverting traffic to a scrubbing center. If the DOTS session flaps due to anycast changes as described above, mitigation may also flap as the DOTS servers sharing the anycast DOTS service address toggles mitigation on detecting DOTS session loss, depending on whether or not the client has configured mitigation on loss of signal (Section 3.3.3).

3.2.5. Signaling Considerations for Network Address Translation

Network address translators (NATs) are expected to be a common feature of DOTS deployments. The middlebox traversal guidelines in [RFC8085] include general NAT considerations that are applicable to DOTS deployments when the signal channel is established over UDP.

Additional DOTS-specific considerations arise when NATs are part of the DOTS architecture. For example, DDoS attack detection behind a NAT will detect attacks against internal addresses. A DOTS client subsequently asked to request mitigation for the attacked scope of addresses cannot reasonably perform the task, due to the lack of externally routable addresses in the mitigation scope.

The following considerations do not cover all possible scenarios but are meant rather to highlight anticipated common issues when signaling through NATs.

3.2.5.1. Direct Provisioning of Internal-to-External Address Mappings

Operators may circumvent the problem of translating internal addresses or prefixes to externally routable mitigation scopes by directly provisioning the mappings of external addresses to internal protected resources on the DOTS client. When the operator requests mitigation scoped for internal addresses, directly or through automated means, the DOTS client looks up the matching external addresses or prefixes and issues a mitigation request scoped to that externally routable information.

When directly provisioning the address mappings, operators must ensure the mappings remain up to date or they risk losing the ability to request accurate mitigation scopes. To that aim, the DOTS client can rely on mechanisms such as [RFC8512] or [RFC7658] to retrieve static explicit mappings. This document does not prescribe the method by which mappings are maintained once they are provisioned on the DOTS client.

3.2.5.2. Resolving Public Mitigation Scope with Port Control Protocol (PCP)

Port Control Protocol (PCP) [RFC6887] may be used to retrieve the external addresses/prefixes and/or port numbers if the NAT function embeds a PCP server.

A DOTS client can use the information retrieved by means of PCP to feed the DOTS protocol(s) messages that will be sent to a DOTS server. These messages will convey the external addresses/prefixes as set by the NAT.

PCP also enables discovery and configuration of the lifetime of port mappings instantiated in intermediate NAT devices. Discovery of port mapping lifetimes can reduce the dependency on heartbeat messages to maintain mappings and, therefore, reduce the load on DOTS servers and the network.

3.2.5.3. Resolving Public Mitigation Scope with Session Traversal Utilities (STUN)

An internal resource, e.g., a web server, can discover its reflexive transport address through a STUN Binding request/response transaction, as described in [RFC8489]. After learning its reflexive transport address from the STUN server, the internal resource can export its reflexive transport address and internal transport address to the DOTS client, thereby enabling the DOTS client to request mitigation with the correct external scope, as depicted in Figure 13. The mechanism for providing the DOTS client with the reflexive transport address and internal transport address is unspecified in this document.

In order to prevent an attacker from modifying the STUN messages in transit, the STUN client and server must use the message-integrity mechanism discussed in Section 9 of [RFC8489] or use STUN over DTLS [RFC7350] or STUN over TLS. If the STUN client is behind a NAT that performs Endpoint-Dependent Mapping [RFC5128], the internal service cannot provide the DOTS client with the reflexive transport address discovered using STUN. The behavior of a NAT between the STUN client and the STUN server could be discovered using the experimental techniques discussed in [RFC5780], but note that there is currently no standardized way for a STUN client to reliably determine if it is behind a NAT that performs Endpoint-Dependent Mapping.

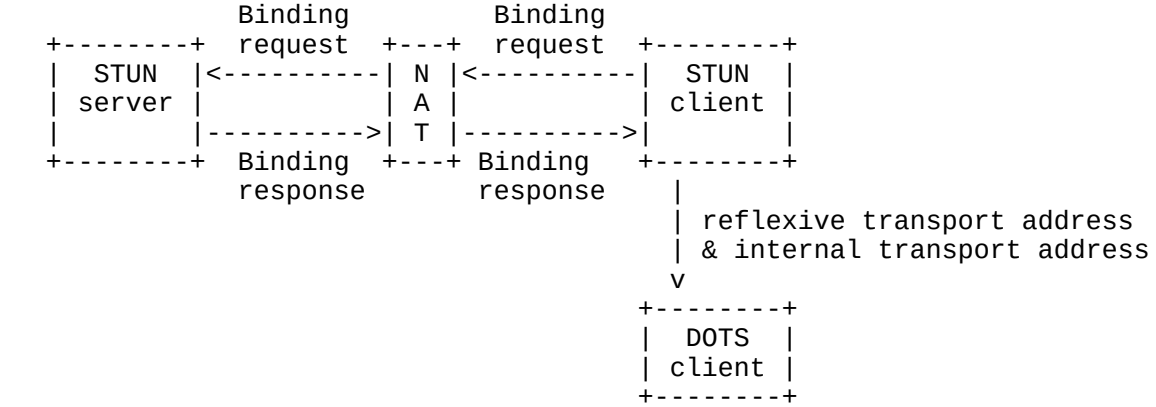


Figure 13: Resolving Mitigation Scope with STUN

3.2.5.4. Resolving Requested Mitigation Scope with DNS

DOTS supports mitigation scoped to DNS names. As discussed in [RFC3235], using DNS names instead of IP addresses potentially avoids the address translation problem, as long as the same domain name is internally and externally resolvable. For example, a detected attack's internal target address can be mapped to a DNS name through

a reverse lookup. The DNS name returned by the reverse lookup can then be provided to the DOTS client as the external scope for mitigation. For the reverse DNS lookup, DNS Security Extensions (DNSSEC) [RFC4033] must be used where the authenticity of response is critical.

3.3. Triggering Requests for Mitigation

[RFC8612] places no limitation on the circumstances in which a DOTS client operator may request mitigation, nor does it demand justification for any mitigation request, thereby reserving operational control over DDoS defense for the domain requesting mitigation. This architecture likewise does not prescribe the network conditions and mechanisms triggering a mitigation request from a DOTS client.

However, considering selected possible mitigation triggers from an architectural perspective offers a model for alternative or unanticipated triggers for DOTS deployments. In all cases, what network conditions merit a mitigation request are at the discretion of the DOTS client operator.

The mitigation request itself is defined by DOTS; however, the interfaces required to trigger the mitigation request in the following scenarios are implementation specific.

3.3.1. Manual Mitigation Request

A DOTS client operator may manually prepare a request for mitigation, including scope and duration, and manually instruct the DOTS client to send the mitigation request to the DOTS server. In context, a manual request is a request directly issued by the operator without automated decision making performed by a device interacting with the DOTS client. Modes of manual mitigation requests include an operator entering a command into a text interface, or directly interacting with a graphical interface to send the request.

An operator might do this, for example, in response to notice of an attack delivered by attack detection equipment or software, and the alerting detector lacks interfaces or is not configured to use available interfaces to translate the alert to a mitigation request automatically.

In a variation of the above scenario, the operator may have preconfigured on the DOTS client mitigation requests for various resources in the operator’s domain. When notified of an attack, the DOTS client operator manually instructs the DOTS client to send the relevant preconfigured mitigation request for the resources under attack.

A further variant involves recursive signaling, as described in Section 3.2.3. The DOTS client in this case is the second half of a DOTS gateway (back-to-back DOTS server and client). As in the previous scenario, the scope and duration of the mitigation request are preexisting but, in this case, are derived from the mitigation request received from a downstream DOTS client by the DOTS server. Assuming the preconditions required by Section 3.2.3 are in place, the DOTS gateway operator may at any time manually request mitigation from an upstream DOTS server, sending a mitigation request derived from the downstream DOTS client’s request.

The motivations for a DOTS client operator to request mitigation manually are not prescribed by this architecture but are expected to include some of the following:

* Notice of an attack delivered via email or alternative messaging

* Notice of an attack delivered via phone call

* Notice of an attack delivered through the interface(s) of networking monitoring software deployed in the operator's domain

* Manual monitoring of network behavior through network monitoring software

3.3.2. Automated Conditional Mitigation Request

Unlike manual mitigation requests, which depend entirely on the DOTS client operator's capacity to react with speed and accuracy to every detected or detectable attack, mitigation requests triggered by detected attack conditions reduce the operational burden on the DOTS client operator and minimize the latency between attack detection and the start of mitigation.

Mitigation requests are triggered in this scenario by operator-specified network conditions. Attack detection is deployment specific and not constrained by this architecture. Similarly, the specifics of a condition are left to the discretion of the operator, though common conditions meriting mitigation include the following:

* Detected attack exceeding a rate in packets per second (pps).

* Detected attack exceeding a rate in bytes per second (bps).

* Detected resource exhaustion in an attack target.

* Detected resource exhaustion in the local domain's mitigator.

* Number of open connections to an attack target.

* Number of attack sources in a given attack.

* Number of active attacks against targets in the operator's domain.

* Conditional detection developed through arbitrary statistical analysis or deep learning techniques.

* Any combination of the above.

When automated conditional mitigation requests are enabled, violations of any of the above conditions, or any additional operator-defined conditions, will trigger a mitigation request from the DOTS client to the DOTS server. The interfaces between the application detecting the condition violation and the DOTS client are implementation specific.

3.3.3. Automated Mitigation on Loss of Signal

To maintain a DOTS signal channel session, the DOTS client and the DOTS server exchange regular but infrequent messages across the signal channel. In the absence of an attack, the probability of message loss in the signaling channel should be extremely low. Under attack conditions, however, some signal loss may be anticipated as attack traffic congests the link, depending on the attack type.

While [RFC8612] specifies the DOTS protocol be robust when signaling under attack conditions, there are nevertheless scenarios in which the DOTS signal is lost in spite of protocol best efforts. To handle such scenarios, a DOTS operator may request one or more mitigations, which are triggered only when the DOTS server ceases receiving DOTS client heartbeats beyond the miss count or interval permitted by the protocol.

The impact of mitigating due to loss of signal in either direction must be considered carefully before enabling it. Attack traffic congesting links is not the only reason why signal could be lost, and as such, mitigation requests triggered by signal channel degradation in either direction may incur unnecessary costs due to scrubbing traffic, adversely impact network performance and operational expense alike.

4. IANA Considerations

This document has no IANA actions.

5. Security Considerations

This section describes identified security considerations for the DOTS architecture.

Security considerations and security requirements discussed in [RFC8612] need to be taken into account.

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 DOTS agents.

Any attacker with the ability to impersonate a legitimate DOTS client or server or, indeed, inject false messages into the stream may potentially trigger/withdraw traffic redirection, trigger/cancel mitigation activities or subvert drop-/accept-lists. From an architectural standpoint, operators MUST ensure conformance to the security requirements defined in Section 2.4 of [RFC8612] to secure data in transit. Similarly, as the received data may contain network topology, telemetry, and threat and mitigation information that could be considered sensitive in certain environments, it SHOULD be protected at rest per required local policy.

DOTS agents MUST perform mutual authentication to ensure authenticity of each other, and DOTS servers MUST verify that the requesting DOTS client is authorized to request mitigation for specific target resources (see Section 2.2.2).

A man-in-the-middle (MITM) attacker can intercept and drop packets, preventing the DOTS peers from receiving some or all of the DOTS messages; automated mitigation on loss of signal can be used as a countermeasure but with risks discussed in Section 3.3.3.

An attacker with control of a DOTS client may negatively influence network traffic by requesting and withdrawing requests for mitigation for particular prefixes, leading to route or DNS flapping. DOTS operators should carefully monitor and audit DOTS clients to detect misbehavior and deter misuse.

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. DOTS servers MUST be given adequate protections in accordance with best current practices for network and host security.

6. References

6.1. 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>.

[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>.

[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>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[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>.

[RFC8612] Mortensen, A., Reddy, T., and R. Moskowitz, "DDoS Open Threat Signaling (DOTS) Requirements", RFC 8612, DOI 10.17487/RFC8612, May 2019, <https://www.rfc-editor.org/info/rfc8612>.

6.2. Informative References

[DOTS-USE-CASES]
Dobbins, R., Migault, D., Moskowitz, R., Teague, N., Xia, L., and K. Nishizuka, "Use cases for DDoS Open Threat Signaling", Work in Progress, Internet-Draft, draft-ietf-dots-use-cases-25, 5 July 2020, <https://tools.ietf.org/html/draft-ietf-dots-use-cases-25>.

[DTLS-PROTOCOL]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-38, 29 May 2020, <https://tools.ietf.org/html/draft-ietf-tls-dtls13-38>.

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

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

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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, <https://www.rfc-editor.org/info/rfc2782>.

[RFC3235] Senie, D., "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC 3235, DOI 10.17487/RFC3235, January 2002, <https://www.rfc-editor.org/info/rfc3235>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,

A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[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, <https://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, <https://www.rfc-editor.org/info/rfc4732>.

[RFC5128] Srisuresh, P., Ford, B., and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", RFC 5128, DOI 10.17487/RFC5128, March 2008, <https://www.rfc-editor.org/info/rfc5128>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, DOI 10.17487/RFC5780, May 2010, <https://www.rfc-editor.org/info/rfc5780>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[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>.

[RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", RFC 7092, DOI 10.17487/RFC7092, December 2013, <https://www.rfc-editor.org/info/rfc7092>.

[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>.

[RFC7350] Petit-Huguenin, M. and G. Salgueiro, "Datagram Transport Layer Security (DTLS) as Transport for Session Traversal Utilities for NAT (STUN)", RFC 7350, DOI 10.17487/RFC7350, August 2014, <https://www.rfc-editor.org/info/rfc7350>.

[RFC7658] Perreault, S., Tsou, T., Sivakumar, S., and T. Taylor, "Deprecation of MIB Module NAT-MIB: Managed Objects for Network Address Translators (NATs)", RFC 7658, DOI 10.17487/RFC7658, October 2015, <https://www.rfc-editor.org/info/rfc7658>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8489] Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and P. Matthews, "Session Traversal Utilities for NAT (STUN)", RFC 8489, DOI 10.17487/RFC8489, February 2020, <https://www.rfc-editor.org/info/rfc8489>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <https://www.rfc-editor.org/info/rfc8512>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8738] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) IP Identifier Validation Extension", RFC 8738, DOI 10.17487/RFC8738, February 2020, <https://www.rfc-editor.org/info/rfc8738>.

[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>.

Acknowledgments

Thanks to Matt Richardson, Roman Danyliw, Frank Xialiang, Roland Dobbins, Wei Pan, Kaname Nishizuka, Jon Shallow, Paul Kyzivat, Warren Kumari, Benjamin Kaduk, and Mohamed Boucadair for their comments and suggestions.

Special thanks to Roman Danyliw for the AD review.

Contributors

Mohamed Boucadair
Orange
mohamed.boucadair@orange.com

Cristopher Gray
Christopher_Gray3@cable.comcast.com

Authors' Addresses

Andrew Mortensen (editor)
Forcepoint
United States of America

Email: andrewmortensen@gmail.com

Tirumaleswar Reddy.K (editor)
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India

Email: kondtir@gmail.com

Flemming Andreasen
Cisco
United States of America

Email: fandreas@cisco.com

Nik Teague
Iron Mountain
United States of America

Email: nteague@ironmountain.co.uk

Rich Compton
Charter

Email: Rich.Compton@charter.com