

I2NSF  
Internet-Draft  
Intended status: Standards Track  
Expires: September 22, 2016

S. Hares  
Huawei  
R. Moskowitz  
HTT Consulting  
March 21, 2016

Secure Session Layer Services  
draft-hares-i2nsf-slss-00.txt

Abstract

Each I2NSF agent and I2NSF client needs to provide application level support for management traffic during periods of DDoS and network security attacks to deal with congestion (burst and/or continuous), high error rates and packet loss due to the attacks, and the inability to utilize a transport protocol (E.g. TCP) due to a specific protocol attack. This application level support needs to be able to select the key management system and provide "chunking" of data (in order to fit in reduced effective MTUs), compression of data (in order to fit into reduced bandwidth), small security envelope (in order to maximize room for management payload), and fragmentation and reassembly at the application layer for those protocols which do not support fragmentation/reassembly (E.g. UDP or SMS). The application layer needs to be able to turn off this features if the system detects these features are no longer needed.

This draft specifies a security session layer services (SSLs) which provide these features in terms of an API, and the component features (interface to key management systems, data compression, chunking of data, secure session envelope (SSE) to send data, and fragmentation and reassembly, and ability to detect existence of attack).

Status of This Memo

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

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

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

This Internet-Draft will expire on September 22, 2016.

#### Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. API for SSLS . . . . .	4
2.1. SSLS socket calls . . . . .	4
2.1.1. KMP related options . . . . .	5
2.1.2. SSE Envelope related options . . . . .	6
2.2. OpenSSL X.509 API calls used . . . . .	7
2.3. HIPv2 API calls used . . . . .	7
2.3.1. HIP Structures . . . . .	7
2.3.2. HIP KMP calls . . . . .	8
3. Data Compression . . . . .	8
4. SSLS Processes . . . . .	8
4.1. Chunking of Data . . . . .	8
4.2. Secure Session Envelope . . . . .	9
4.3. Application Packet Fragmentation and Reassembly . . . . .	10
4.4. Proprietary Plugins: Detect Conditions + Select Transport . . . . .	13
5. IANA Considerations . . . . .	13
6. Security Considerations . . . . .	13
7. Acknowledgements . . . . .	14
8. References . . . . .	14
8.1. Normative References . . . . .	14
8.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

#### 1. Introduction

Each I2NSF agent and I2NSF client needs to provide application level support for management traffic during periods of DDoS and network security attacks to deal with congestion (burst and/or continuous), high error rates and packet loss due to the attacks, and the

inability to utilize a transport protocol (E.g. TCP) due to a specific protocol attack. Some of the services the I2NSF controller must provide during these periods of DDoS or network security attacks are:

- o receiving information regarding DDoS Threats from DOTS systems,
- o Changing policy on vNSF and NSF devices during these periods,
- o exchanging information with user security applications using I2NSF to obtain information from the controller,
- o Aid the I2NSF reporting of attacks with the the CERT (MILE) either by providing data or sendign the report
- o and manages network connectivity of devices out of compliance (SACM) .

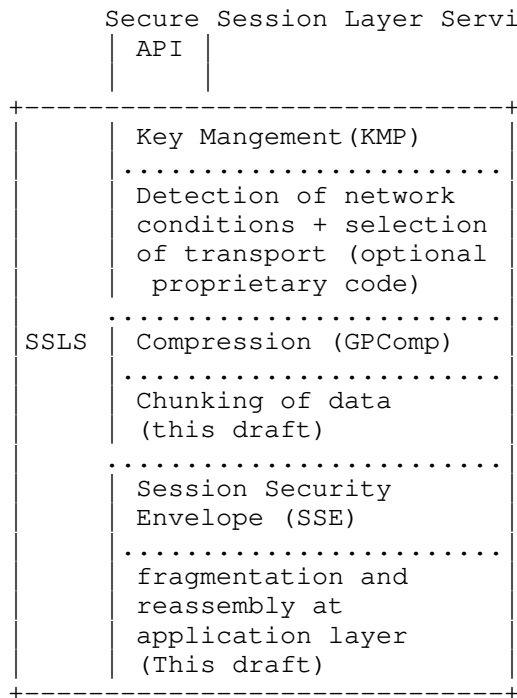
This application level support for I2NSF client-agent communication needs to be able to select the key management system and provide "chunking" of data (in order to fit in reduced effective MTUs), compression of data (in order to fit into reduced bandwidth), small security envelope )in order to maximize room for mangement payload), and fragmentation and reassembly at the application layer for those protocols which do not support fragmentation/reassembly (E.g. UDP or SMS). The application layer needs to be able to turn off this features if the system detects these features are no longer needed.

This draft specifies a security session layer (SSL) which provides these features in terms of:

- o an API for the layer (section 2)
- o interface to key management system (section 3),
- o data compression (section 4)
- o chunking of data (section 5)
- o secure envelope (section 6),
- o fragmentation and reassembly (section 7),
- o detection of network conditions that require this service (section 8) .

A diagram of the SSLS with these process is in figure 1.

The API for this SSLS allows the application to select the types of key management, and the different types of services (data compression, chunking of data, secure e)



## 2. API for SSLS

### 2.1. SSLS socket calls

The SSLS uses socket calls to set up the application session layer. The calls are shown below.

```
s = int socket(int domain, int type, int protocol)
```

where:

domain: AF\_INET and AF\_INET6 supported

type: SOCK\_SSLS

desired protocol: Transport protocol (TCP (6), UDP (6), SCTP (132)), SMS (xx)

```
int setsockopt(int sockfd, int level, int optname,
               const void *optval, socklen_t optlen);

int getsockopt(int sockfd, int level, int optname,
               const void *optval, socklen_t optlen);

where:
sockfd:      # socket file descriptor
optname:     # option name (see below)
optval;      # points to *sse_transport structure;
optlen;      # length of option

optnam:
SSLS_AUTH_PRIV ]1]
SSLS_AES_MODE[2]
SSLS_ALGS[3]
SSLS_SSE [4]
```

Where the opt\_val structure are define in the figure below.

Figure 2

#### 2.1.1. KMP related options

Security Keying structures for:  
 SSLS\_AUTH\_PRIV, SSLS\_AES\_MODE, SSLS\_ALGS  
 options of setsockopt, getsockopt

```
#struture for SSL_AUTH-PRIV optval
struct *ssls_auth_priv_opts {
    *ssls-x509-auth [SSLS-X509-LIMIT]
}

#SSL-X509-limit
typedef struct ssls-x509-auth {
    const char name;
    void *x509-cert; #cert struture by API
}

#structure for SSL_AES_MODE optval
struct *ssls_aes_mode_opts {
... IKEV2 options # openikev2 API
... HIPv2 options # HIPv2 API
                                                    #[RFC6317 + HIPv2]
struct ssls_algs_opts;
}

#compression options
struct *ssls_algs_opts {
    boolean gpcomp-kmp; # computed with keys
enum gmcomp-type; #
}
```

figure 3: setsockopt structure  
 for KMP related optins

#### 2.1.2. SSE Envelope related options

```

Security Session Envelope Related options
#structure for SSL_SSE optval
# SPI - is generated by KMP
# SSE - sequence number - by SSE
# Flags = Fragment (5 bits [0-5],

struct *ssls_sse_opts {
    int nt_sockfd;      # new transport socket
    int *protocol;      # transport protocol for SSLS SSE
                        # can choose from (1-n )
    int *known_ports    # known ports
    int chunk-size;     # chunk size
    int frag-size;      # fragment size
                        # greater than 0 means fragment]
    int SSEs-at-once    # number of SSEs sent at once
    enum SSE_size;      # (compact, large, extreme)
    enum SSE-FLAG;      # compression flags
};

```

Figure 4

## 2.2. OpenSSL X.509 API calls used

TBD

## 2.3. HIPv2 API calls used

(API calls will be added later based on HIP [RFC6317] upgraded to HIPv2.

### 2.3.1. HIP Structures

```

struct addrinfo {
    int      ai_flags;          /* e.g., AI_CANONNAME */
    int      ai_family;        /* e.g., AF_HIP */
    int      ai_socktype;      /* e.g., SOCK_STREAM */
    int      ai_protocol;      /* 0 or IPPROTO_HIP */
    socklen_t ai_addrlen;      /* size of *ai_addr */
    struct   sockaddr *ai_addr; /* sockaddr_hip */
    char     *ai_canonname;     /* canon. name of the host */
    struct   addrinfo *ai_next; /* next endpoint */
    int      ai_eflags;        /* RFC 5014 extension */
};

```

### 2.3.2. HIP KMP calls

```
#HIP uses
# #include <netdb.h>
int getaddrinfo(const char *nodename,
                const char *servname,
                const struct addrinfo *hints,
                struct addrinfo **res)
void free_addrinfo(struct addrinfo *res)
```

Figure 3

## 3. Data Compression

The first step in making the application data easier to send through the network is to compress the data. The data compression algorithm is defined in draft-moskowitz-gpcomp-00.txt. The result of the compressed data is handed to the chunking function.

The user can disable or enable the compression function by setting SSE-DATA types to be one of the following:

- o SSLS compress only - set compression, [1]
- o SSLS compression and fragmentation [3],

Setting this flag to:

- o no compression or fragmentation [0],
- o SSLS to fragmentation only [2]

will skip the data compression step.

## 4. SSLS Processes

### 4.1. Chunking of Data

The process that "chunks" data breaks down the application stream after the compression process. If the compression process has compressed the data, the chunking process will chunk compressed data. If the user has requested no compression, this chunking process will chunk uncompressed data. The size of chunks of data the SSLS process creates to encapsulate in the secure session envelope (SSE) is specified on SSL\_SSE setsockopt call.

The secure session envelope must be bigger than the chunk.



If the SSE is using TCP or STCP, that assembles the application flow into a byte stream, then the SSE packages will contain a chunk within the secure session envelope.

If Transports that do not fragment and re-assembly are being specified, the SSL will support application layer fragmentation and reassembly. (see the fragmentation section below)

#### 4.2. Secure Session Envelope

The Secure Session Envelope (SSE) creates a secure envelope using the SPI created by the key management and running over the transport selected by the user. The SSE has three forms: compact, Large, Extreme. The SSE compact form is below in figure x. SSL defines 4 bytes of the reserved field in the FLAGS field. See [I-D.moskowitz-sse] for details on secure session envelope sizes and formats.

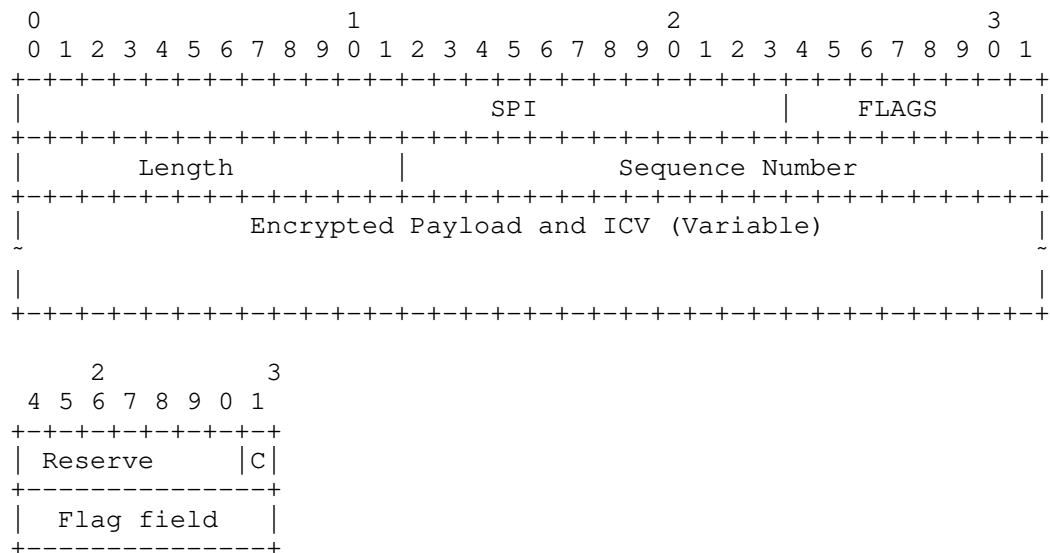
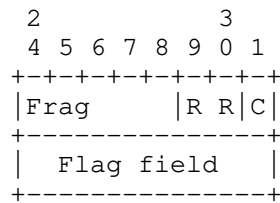


Figure 5 - Compact format of SSE

The SSLS utilizes 6 bits of the 8 bit flag in order to provide fragmentation and reassembly checks when the SSE gets fragmented into multiple transport packets. Each time the SSE fragments the packet to fit in the transport, it increments the fragment count in bits [24-28]. The bits for the flag word shown in figure 6.



Flag work in SSE header

Bits [4-8] - 1-30 bit value for the fragment number  
           0 - no fragmentation  
           31 - indicates an fragmentation ACK response  
 Bits 5-6 - reserved  
 Bit      7 - compression

Figure 6 - SSLS redefined SSE Flag byte

#### 4.3. Application Packet Fragmentation and Reassembly

SSE's secure envelope may be passed over UDP to avoid transport-level security attacks. Alternatively SSE's secure transport may go over the extremely limited SMS fabric so that some security management information gets through. In both cases, the user (or the "detection log") can select the transport and fragmentation.

If fragmentation is turned on, the individual SSE envelopes will track the IP messages the SSE envelope is broken into by placing the fragment number in the lowest 5 bits of the SSE Flag byte [24-28]. The SSE process receiving the traffic will send back an acknowledge SSE packet [Flag value in bits 0-4 is 0x1F or 31] within 30 bit map of sequences acked [1-30] in first 4 bits of SSE data. It is anticipate that the fragmentation process will attempt to bundle some acks.

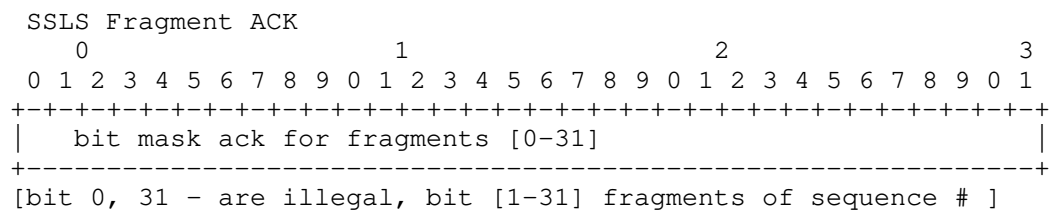
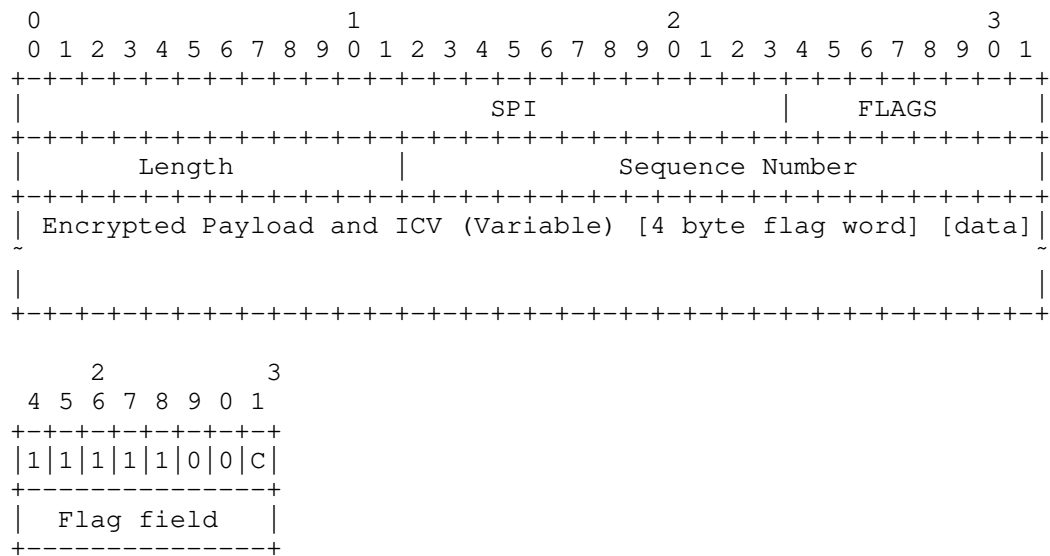


Figure 7 - SSLS ACK flag filed and first 4 bytes of payload

### An example Fragmentation and ACK exchange

```

SSLs-process-1-----IP/SMS-----SSLs Process-2
[E.g. I2NSF Client -----I2NSF Agent]

SSE-packet (SPI,(flags(fragment=1,C=1),
             length, seq 1, data )---->

SSE-packet (SPI,(flags(fragment=2,C=1),
             length, seq 1, data )---->

SSE-packet (SPI,(flags(fragment=3,C=1),
             length, seq 1, data )---->

SSE-packet (SPI,(flags(fragment=1,C=1),
             length, seq 2, data )---->

SSE-packet (SPI,(flags(fragment=2,C=1),
             length, seq 2, data )---->
             <--SSE-packet (SPI)(flags fragment=31,C=1)
             length, seq1,[ack-fragment 1,2])
             <--SSE-packet (SPI)(flags fragment=32,C=1)
             length, seq2,[ack-fragment1,2]

SSE-packet (SPI,(flags(fragment=3,C=1),
             length, seq 1, data )---->
             <--SSE-packet (SPI)(flags fragment=31,C=1)
             length, seq1,[ack-fragment 3])

```

Below is a set of pseudo call for the calls to socket

```

pseudo
struct sse_opts = {};
optlen=size(sse_opts);
optname= SSLs_SSE; #4
s = int socket(int domain, int type, int protocol)
errno = int setsockopt(sockfd,level,optname,
                      void struct *sse_opts,optlen);

```

Errors: (Exact ERNOS added later)

- protocol not support
- error in known ports
- error in chunk\_size
- error in fragment size
- error in SSE-at-once
- error - unsupported SSE
- error in compression flags

[Add read-write to socket ]

The SSE window size for fragmentation is 30 IP fragments or 30 SMS fragments per SSE chunk. The SSE process SHOULD assign the SSE fragments in order if possible. The SSE process will send an error response to the SSE if the data chunk does not fit in 30 IP/SM fragments.

If the SSE transmitting process has not received an acknowledgement for all IP fragments for a particular SSE envelope (identified by sequence number) with a SSE-retransmit-time, it will retransmit the unacknowledged fragments.

Several SSE envelopes may be sent with fragmentation at once. The user signals the number sent at once with multiple SSE with fragment variable on the options. If fragmentation is selected, each of these SSE envelopes may need to track up to 30 IP fragments.

#### 4.4. Proprietary Plugins: Detect Conditions + Select Transport

The SSL process allows two proprietary plugins:

1. Plugin to detect error conditions which require SSLS services which include:
  - \* High levels of end-to-end congestion,
  - \* High levels of error and loss,
  - \* Input from IDS/IPS that detects problems
  - \* Signals from other I2NSF applications
2. Proprietary actions may select transport based on input from other standardized security services (DOTS, CERT, MILE) or proprietary services.

Prototype code will provide instances to show plugin values.

#### 5. IANA Considerations

TBD

#### 6. Security Considerations

The SSLS shares the following security considerations with the SSE Technology:

- o As SSE uses an AEAD block cipher, it is vulnerable to attack if a sequence number is reused for a given key. Thus implementations

of SSE MUST provide for rekeying prior to Sequence Number rollover. An implementation should never assume that for a given context, the sequence number space will never be exhausted. Key Management Protocols like IKEv2 [RFC7296] or HIP [RFC7401] could be used to provide for rekeying management. The KMP SHOULD not create a network layer fate-sharing limitation.

- o As any security protocol can be used for a resource exhaustion attack, implementations should consider methods to mitigate flooding attacks of messages with valid SPIs but invalid content. Even with the ICV check, resources are still consumed to validate the ICV.
- o SSE makes no attempt to recommend the ICV length. For constrained network implementations, other sources should guide the implementation as to ICV length selection. The ICV length selection SHOULD be the responsibility of the KMP.
- o As with any layered security protocol, SSE makes no claims of protecting lower or higher processes in the communication stack. Each layer's risks and liabilities need be addressed at that level.

## 7. Acknowledgements

The authors would like to thank Frank (Liang) Xia for his comments and suggestions on this draft.

## 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.hares-i2nsf-mgtflow-reqs]  
Hares, S., "I2NSF Data Flow Requirements", draft-hares-i2nsf-mgtflow-reqs-00 (work in progress), March 2016.

[I-D.moskowitz-sse]  
Moskowitz, R., Faynberg, I., Lu, H., Hares, S., and P. Giacomin, "Session Security Envelope", draft-moskowitz-sse-02 (work in progress), February 2016.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6317] Komu, M. and T. Henderson, "Basic Socket Interface Extensions for the Host Identity Protocol (HIP)", RFC 6317, DOI 10.17487/RFC6317, July 2011, <<http://www.rfc-editor.org/info/rfc6317>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<http://www.rfc-editor.org/info/rfc7401>>.

## Authors' Addresses

Susan Hares  
Huawei  
Saline  
US

Email: [shares@ndzh.com](mailto:shares@ndzh.com)

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 48237  
USA

Phone: +1-248-968-9809  
Email: [rgm@htt-consult.com](mailto:rgm@htt-consult.com)

DOTS  
Internet-Draft  
Intended status: Informational  
Expires: September 19, 2016

A. Mortensen  
Arbor Networks, Inc.  
R. Moskowitz  
HTT Consulting  
T. Reddy  
Cisco Systems, Inc.  
March 18, 2016

Distributed Denial of Service (DDoS) Open Threat Signaling Requirements  
draft-ietf-dots-requirements-01

Abstract

This document defines the requirements for the Distributed Denial of Service (DDoS) Open Threat Signaling (DOTS) protocols coordinating attack response against DDoS attacks.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 19, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of



the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Context and Motivation . . . . .	2
1.2. Terminology . . . . .	3
2. Requirements . . . . .	5
2.1. General Requirements . . . . .	6
2.2. Operational Requirements . . . . .	7
2.3. Data Channel Requirements . . . . .	10
2.4. Security requirements . . . . .	11
3. Congestion Control Considerations . . . . .	12
4. Security Considerations . . . . .	12
5. Contributors . . . . .	12
6. Acknowledgments . . . . .	12
7. Change Log . . . . .	13
7.1. 01 revision . . . . .	13
7.2. 00 revision . . . . .	13
7.3. Initial revision . . . . .	13
8. References . . . . .	14
8.1. Normative References . . . . .	14
8.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

### 1.1. Context and Motivation

Distributed Denial of Service (DDoS) attacks continue to plague networks around the globe, from Tier-1 service providers on down to enterprises and small businesses. Attack scale and frequency similarly have continued to increase, in part as a result of software vulnerabilities leading to reflection and amplification attacks. Once staggering attack traffic volume is now the norm, and the impact of larger-scale attacks attract the attention of international press agencies.

The greater impact of contemporary DDoS attacks has led to increased focus on coordinated attack response. Many institutions and enterprises lack the resources or expertise to operate on-premise attack prevention solutions themselves, or simply find themselves constrained by local bandwidth limitations. To address such gaps, security service providers have begun to offer on-demand traffic scrubbing services. Each service offers its own interface for subscribers to request attack mitigation, tying subscribers to proprietary implementations while also limiting the subset of network

elements capable of participating in the attack response. As a result of incompatibility across services, attack responses may be fragmentary or otherwise incomplete, leaving key players in the attack path unable to assist in the defense.

The lack of a common method to coordinate a real-time response among involved actors and network domains inhibits the speed and effectiveness of DDoS attack mitigation. This document describes the required characteristics of DOTS protocols that would mitigate contemporary DDoS attack impact and lead to more efficient defensive strategies.

DOTS is less concerned with the form of defensive action than with communicating the need for that action. DOTS supplements calls for help with pertinent details about the detected attack, allowing entities participating in DOTS to form ad hoc, adaptive alliances against DDoS attacks as described in the DOTS use cases [I-D.ietf-dots-use-cases]. The requirements in this document are derived from those use cases.

## 1.2. Terminology

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].

This document adopts the following terms:

**DDoS:** A distributed denial-of-service attack, in which of traffic originating from multiple sources are directed at a target on a network. DDoS attacks are intended to cause a negative impact on the availability of servers, services, applications, and/or other functionality of an attack target. Denial-of-service considerations are discussed in detail in [RFC4732].

**DDoS attack target:** A networked server, network service or application that is the focus of a DDoS attack.

**DDoS attack telemetry:** Collected network traffic characteristics defining the nature of a DDoS attack. This document makes no assumptions regarding telemetry collection methodology.

**Countermeasure:** An action or set of actions taken to recognize and filter out DDoS attack traffic while passing legitimate traffic to the attack target.

**Mitigation:** A set of countermeasures enforced against traffic destined for the target or targets of a detected or reported DDoS

attack, where countermeasure enforcement is managed by an entity in the network path between attack sources and the attack target. Mitigation methodology is out of scope for this document.

**Mitigator:** An entity, typically a network element, capable of performing mitigation of a detected or reported DDoS attack. For the purposes of this document, this entity is a black box capable of mitigation, making no assumptions about availability or design of countermeasures, nor about the programmable interface between this entity and other network elements. The mitigator and DOTS server are assumed to belong to the same administrative entity.

**DOTS client:** A DOTS-aware software module responsible for requesting attack response coordination with other DOTS-aware elements.

**DOTS server:** A DOTS-aware software module handling and responding to messages from DOTS clients. The DOTS server MAY enable mitigation on behalf of the DOTS client, if requested, by communicating the DOTS client's request to the mitigator and relaying any mitigator feedback to the requesting DOTS client. A DOTS server may also be a mitigator.

**DOTS relay:** A DOTS-aware software module positioned between a DOTS server and a DOTS client in the signaling path. A DOTS relay receives messages from a DOTS client and relays them to a DOTS server, and similarly passes messages from the DOTS server to the DOTS client. A DOTS relay acts as a proxy or bridge between stateful and stateless transport signaling, and may also aggregate signaling from multiple downstream DOTS clients into a single session with an upstream DOTS server or DOTS relay.

**DOTS agents:** Any DOTS functional element, including DOTS clients, DOTS servers and DOTS relays.

**Signal channel:** A bidirectional, mutually authenticated communication layer between DOTS agents characterized by resilience even in conditions leading to severe packet loss, such as a volumetric DDoS attack causing network congestion.

**DOTS signal:** A concise authenticated status/control message transmitted between DOTS agents, used to indicate client's need for mitigation, as well as to convey the status of any requested mitigation.

**Heartbeat:** A message transmitted between DOTS agents over the signal channel, used as a keep-alive and to measure peer health.

**Client signal:** A message sent from a DOTS client to a DOTS server or DOTS relay over the signal channel, indicating the DOTS client's need for mitigation, as well as the scope of any requested mitigation, optionally including additional attack details to supplement server-initiated mitigation.

**Server signal:** A message sent from a DOTS server to a DOTS client or DOTS relay over the signal channel. Note that a server signal is not a response to client signal, but a DOTS server-initiated status message sent to DOTS clients with which the server has established signaling sessions, containing information about the status of DOTS client-requested mitigation and its efficacy.

**Data channel:** A secure communication layer between DOTS clients and DOTS servers used for infrequent bulk exchange of data not easily or appropriately communicated through the signal channel under attack conditions.

**Blacklist:** A list of addresses, prefixes and/or other identifiers indicating sources from which traffic should be blocked, regardless of traffic content.

**Whitelist:** A list of addresses, prefixes and/or other identifiers from indicating sources from which traffic should always be allowed, regardless of contradictory data gleaned in a detected attack.

**Multi-homed DOTS client:** A DOTS client exchanging messages with multiple DOTS servers, each in a separate administrative domain.

## 2. Requirements

This section describes the required features and characteristics of the DOTS protocols.

DOTS is an advisory protocol. An active DDoS attack against the entity controlling the DOTS client need not be present before establishing DOTS communication between DOTS agents. Indeed, establishing a relationship with peer DOTS agents during nominal network conditions provides the foundation for more rapid attack response against future attacks, as all interactions setting up DOTS, including any business or service level agreements, are already complete.

DOTS must at a minimum make it possible for a DOTS client to request a DOTS server's aid in mounting a coordinated defense against a detected attack, signaling inter- or intra-domain as requested by local operators. DOTS clients should similarly be able to withdraw

aid requests. DOTS requires no justification from DOTS clients for requests for help, nor must DOTS clients justify withdrawing help requests: the decision is local to the entity owning the DOTS clients. Regular feedback between DOTS clients and DOTS server supplement the defensive alliance by maintaining a common understanding of DOTS peer health and activity. Bidirectional communication between DOTS clients and DOTS servers is therefore critical.

Yet DOTS must also work with a set of competing operational goals. On the one hand, the protocol must be resilient under extremely hostile network conditions, providing continued contact between DOTS agents even as attack traffic saturates the link. Such resiliency may be developed several ways, but characteristics such as small message size, asynchronous, redundant message delivery and minimal connection overhead (when possible given local network policy) will tend to contribute to the robustness demanded by a viable DOTS protocol. Operators of peer DOTS-enabled domains may enable quality- or class-of-service traffic tagging to increase the probability of successful DOTS signal delivery, but DOTS requires no such policies be in place. The DOTS solution indeed must be viable especially in their absence.

On the other hand, DOTS must include protections ensuring message confidentiality, integrity and authenticity to keep the protocol from becoming another vector for the very attacks it's meant to help fight off. DOTS clients must be able to authenticate DOTS servers, and vice versa, for DOTS to operate safely, meaning the DOTS agents must have a way to negotiate and agree upon the terms of protocol security. Attacks against the transport protocol should not offer a means of attack against the message confidentiality, integrity and authenticity.

The DOTS server and client must also have some common method of defining the scope of any mitigation performed by the mitigator, as well as making adjustments to other commonly configurable features, such as listen ports, exchanging black- and white-lists, and so on.

Finally, DOTS should provide sufficient extensibility to meet local, vendor or future needs in coordinated attack defense, although this consideration is necessarily superseded by the other operational requirements.

## 2.1. General Requirements

GEN-001 Extensibility: Protocols and data models developed as part of DOTS MUST be extensible in order to keep DOTS adaptable to

operational and proprietary DDoS defenses. Future extensions MUST be backward compatible.

GEN-002 Resilience and Robustness: The signaling protocol MUST be designed to maximize the probability of signal delivery even under the severely constrained network conditions imposed by particular attack traffic. The protocol MUST be resilient, that is, continue operating despite message loss and out-of-order or redundant signal delivery.

GEN-003 Bidirectionality: To support peer health detection, to maintain an open signal channel, and to increase the probability of signal delivery during attack, the signal channel MUST be bidirectional, with client and server transmitting signals to each other at regular intervals, regardless of any client request for mitigation. Unidirectional messages MUST be supported within the bidirectional signal channel to allow for unsolicited message delivery, enabling asynchronous notifications between agents.

GEN-004 Sub-MTU Message Size: To avoid message fragmentation and the consequently decreased probability of message delivery, signaling protocol message size MUST be kept under signaling path Maximum Transmission Unit (MTU), including the byte overhead of any encapsulation, transport headers, and transport- or message-level security.

GEN-005 Bulk Data Exchange: Infrequent bulk data exchange between DOTS agents can also significantly augment attack response coordination, permitting such tasks as population of black- or white-listed source addresses; address or prefix group aliasing; exchange of incident reports; and other hinting or configuration supplementing attack response.

As the resilience requirements for the DOTS signal channel mandate small signal message size, a separate, secure data channel utilizing an established reliable transport protocol MUST be used for bulk data exchange.

## 2.2. Operational Requirements

OP-001 Use of Common Transport Protocols: DOTS MUST operate over common widely deployed and standardized transport protocols. While the User Datagram Protocol (UDP) [RFC0768] SHOULD be used for the signal channel, the Transmission Control Protocol (TCP) [RFC0793] MAY be used if necessary due to network policy or middlebox capabilities or configurations. The data channel MUST use TCP; see Section 2.3 below.

OP-002 Session Health Monitoring: Peer DOTS agents MUST regularly send heartbeats to each other after mutual authentication in order to keep the DOTS session open. A session MUST be considered active until a DOTS agent explicitly ends the session, or either DOTS agent fails to receive heartbeats from the other after a mutually negotiated timeout period has elapsed.

OP-003 Session Redirection: In order to increase DOTS operational flexibility and scalability, DOTS servers SHOULD be able to redirect DOTS clients to another DOTS server or relay at any time. Due to the decreased probability of DOTS server signal delivery due to link congestion, it is RECOMMENDED DOTS servers avoid redirecting while mitigation is enabled during an active attack against a target in the DOTS client's domain. Either the DOTS servers have to fate-share the security state, the client MUST have separate security state with each potential redirectable server, or be able to negotiate new state as part of redirection.

OP-004 Mitigation Status: DOTS MUST provide a means to report the status of an action requested by a DOTS client. In particular, DOTS clients MUST be able to request or withdraw a request for mitigation from the DOTS server. The DOTS server MUST acknowledge a DOTS client's request to withdraw from coordinated attack response in subsequent signals, and MUST cease mitigation activity as quickly as possible. However, a DOTS client rapidly toggling active mitigation may result in undesirable side-effects for the network path, such as route or DNS [RFC1034] flapping. A DOTS server therefore MAY continue mitigating for a mutually negotiated period after receiving the DOTS client's request to stop.

A server MAY refuse to engage in coordinated attack response with a client. To make the status of a client's request clear, the server MUST indicate in server signals whether client-initiated mitigation is active. When a client-initiated mitigation is active, and threat handling details such as mitigation scope and statistics are available to the server, the server SHOULD include those details in server signals sent to the client. DOTS clients SHOULD take mitigation statistics into account when deciding whether to request the DOTS server cease mitigation.

OP-005 Mitigation Lifetime: A DOTS client SHOULD indicate the desired lifetime of any mitigation requested from the DOTS server. As DDoS attack duration is unpredictable, the DOTS client SHOULD be able to extend mitigation lifetime with periodic renewed requests for help. When the mitigation lifetime comes to an end, the DOTS server SHOULD delay session termination for a protocol-defined grace period to allow for delivery of delayed mitigation

renewals over the signal channel. After the grace period elapses, the DOTS server MAY terminate the session at any time.

If a DOTS client does not include a mitigation lifetime in requests for help sent to the DOTS server, the DOTS server will use a reasonable default as defined by the protocol. As above, the DOTS client MAY extend a current mitigation request's lifetime trivially with renewed requests for help.

A DOTS client MAY also request an indefinite mitigation lifetime, enabling architectures in which the mitigator is always in the traffic path to the resources for which the DOTS client is requesting protection. DOTS servers MAY refuse such requests for any reason. The reasons themselves are not in scope.

OP-006 Mitigation Scope: DOTS clients MUST indicate the desired address or prefix space coverage of any mitigation, for example by using Classless Internet Domain Routing (CIDR) [RFC1518], [RFC1519] prefixes, [RFC2373] for IPv6 [RFC2460] prefixes, the length/prefix convention established in the Border Gateway Protocol (BGP) [RFC4271], SIP URIs [RFC3261], E.164 numbers, DNS names, or by a prefix group alias agreed upon with the server through the data channel.

If there is additional information available narrowing the scope of any requested attack response, such as targeted port range, protocol, or service, DOTS clients SHOULD include that information in client signals. DOTS clients MAY also include additional attack details. Such supplemental information is OPTIONAL, and DOTS servers MAY ignore it when enabling countermeasures on the mitigator.

As an active attack evolves, clients MUST be able to adjust as necessary the scope of requested mitigation by refining the address space requiring intervention.

OP-007 Mitigation Efficacy: When a mitigation request by a DOTS client is active, DOTS clients SHOULD transmit a metric of perceived mitigation efficacy to the DOTS server, per "Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services" in [I-D.ietf-dots-use-cases]. DOTS servers MAY use the efficacy metric to adjust countermeasures activated on a mitigator on behalf of a DOTS client.

OP-008 Conflict Detection and Notification: Multiple DOTS clients controlled by a single administrative entity may send conflicting mitigation requests for pool of protected resources, as a result of misconfiguration, operator error, or compromised DOTS clients.



DOTS servers attempting to honor conflicting requests may flap network route or DNS information, degrading the networks attempting to participate in attack response with the DOTS clients. DOTS servers SHALL detect such conflicting requests, and SHALL notify the DOTS clients in conflict. The notification SHOULD indicate the nature and scope of the conflict, for example, the overlapping prefix range in a conflicting mitigation request.

OP-009: Lookup Caching: DOTS agents SHOULD cache resolved names, PKI validation chains, and similarly queried data as necessary. Network-based lookups and validation may be inhibited or unavailable during an active attack due to link congestion. For example, DOTS agents SHOULD cache resolved names and addresses of peer DOTS agents, and SHOULD refer to those agents by IPv4 [RFC0791] or IPv6 address for all communications following initial name resolution.

OP-010: Network Address Translator Traversal: The DOTS protocol MUST operate over networks in which Network Address Translation (NAT) is deployed. As UDP is the recommended transport for DOTS, all considerations in "Middlebox Traversal Guidelines" in [RFC5405] apply to DOTS. Regardless of transport, DOTS protocols MUST follow established best common practices (BCPs) for NAT traversal.

### 2.3. Data Channel Requirements

The data channel is intended to be used for bulk data exchanges between DOTS agents. Unlike the signal channel, which must operate nominally even when confronted with despite signal degradation due to packet loss, the data channel is not expected to be constructed to deal with attack conditions. As the primary function of the data channel is data exchange, a reliable transport is required in order for DOTS agents to detect data delivery success or failure.

The data channel must be extensible. We anticipate the data channel will be used for such purposes as configuration or resource discovery. For example, a DOTS client may submit to the DOTS server a collection of prefixes it wants to refer to by alias when requesting mitigation, to which the server would respond with a success status and the new prefix group alias, or an error status and message in the event the DOTS client's data channel request failed. The transactional nature of such data exchanges suggests a separate set of requirements for the data channel, while the potentially sensitive content sent between DOTS agents requires extra precautions to ensure data privacy and authenticity.

DATA-001 Reliable transport: Transmissions over the data channel MUST be transactional, requiring reliable, in-order packet delivery.

DATA-002 Data privacy and integrity: Transmissions over the data channel is likely to contain operationally or privacy-sensitive information or instructions from the remote DOTS agent. Theft or modification of data channel transmissions could lead to information leaks or malicious transactions on behalf of the sending agent (see Section 4 below). Consequently data sent over the data channel MUST be encrypted and authenticated using current industry best practices. DOTS servers and relays MUST enable means to prevent leaking operationally or privacy-sensitive data. Although administrative entities participating in DOTS may detail what data may be revealed to third-party DOTS agents, such considerations are not in scope for this document.

DATA-003 Black- and whitelist management: DOTS servers SHOULD provide methods for DOTS clients to manage black- and white-lists of source addresses of traffic destined for addresses belonging to a client.

For example, a DOTS client should be able to create a black- or whitelist entry; retrieve a list of current entries from either list; update the content of either list; and delete entries as necessary.

How the DOTS server determines client ownership of address space is not in scope.

## 2.4. Security requirements

DOTS must operate within a particularly strict security context, as an insufficiently protected signal or data channel may be subject to abuse, enabling or supplementing the very attacks DOTS purports to mitigate.

SEC-001 Peer Mutual Authentication: DOTS agents MUST authenticate each other before a DOTS session is considered valid. The method of authentication is not specified, but should follow current industry best practices with respect to any cryptographic mechanisms to authenticate the remote peer.

SEC-002 Message Confidentiality, Integrity and Authenticity: DOTS protocols MUST take steps to protect the confidentiality, integrity and authenticity of messages sent between client and server. While specific transport- and message-level security

options are not specified, the protocols MUST follow current industry best practices for encryption and message authentication.

In order for DOTS protocols to remain secure despite advancements in cryptanalysis and traffic analysis, DOTS agents MUST be able to negotiate the terms and mechanisms of protocol security, subject to the interoperability and signal message size requirements above.

SEC-003 Message Replay Protection: In order to prevent a passive attacker from capturing and replaying old messages, DOTS protocols MUST provide a method for replay detection.

### 3. Congestion Control Considerations

The DOTS signal channel will not contribute measurably to link congestion, as the protocol's transmission rate will be negligible regardless of network conditions. Bulk data transfers are performed over the data channel, which should use a reliable transport with built-in congestion control mechanisms, such as TCP.

### 4. Security Considerations

DOTS is at risk from three primary attacks:

- o DOTS agent impersonation
- o Traffic injection
- o Signaling blocking

The DOTS protocol MUST be designed for minimal data transfer to address the blocking risk. Impersonation and traffic injection mitigation can be managed through current secure communications best practices. See Section 2.4 above for a detailed discussion.

### 5. Contributors

Med Boucadair

### 6. Acknowledgments

Thanks to Roman Danyliw and Matt Richardson for careful reading and feedback.

## 7. Change Log

### 7.1. 01 revision

2016-03-21

- o Reconciled terminology with -00 revision of [I-D.ietf-dots-use-cases].
- o Terminology clarification based on working group feedback.
- o Moved security-related requirements to separate section.
- o Made resilience/robustness primary general requirement to align with charter.
- o Clarified support for unidirectional communication within the bidirection signal channel.
- o Added proposed operational requirement to support session redirection.
- o Added proposed operational requirement to support conflict notification.
- o Added proposed operational requirement to support mitigation lifetime in mitigation requests.
- o Added proposed operational requirement to support mitigation efficacy reporting from DOTS clients.
- o Added proposed operational requirement to cache lookups of all kinds.
- o Added proposed operational requirement regarding NAT traversal.
- o Removed redundant mutual authentication requirement from data channel requirements.

### 7.2. 00 revision

2015-10-15

### 7.3. Initial revision

2015-09-24 Andrew Mortensen

## 8. References

### 8.1. Normative References

- [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>>.
- [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>>.
- [RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, DOI 10.17487/RFC5405, November 2008, <<http://www.rfc-editor.org/info/rfc5405>>.

### 8.2. Informative References

- [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.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [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>>.
- [RFC1518] Rekhter, Y. and T. Li, "An Architecture for IP Address Allocation with CIDR", RFC 1518, DOI 10.17487/RFC1518, September 1993, <<http://www.rfc-editor.org/info/rfc1518>>.
- [RFC1519] Fuller, V., Li, T., Yu, J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, DOI 10.17487/RFC1519, September 1993, <<http://www.rfc-editor.org/info/rfc1519>>.

- [RFC2373] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, DOI 10.17487/RFC2373, July 1998, <<http://www.rfc-editor.org/info/rfc2373>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [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, <<http://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, <<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>>.

#### Authors' Addresses

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

Email: [amortensen@arbor.net](mailto:amortensen@arbor.net)

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 42837  
United States

Email: [rgm@htt-consult.com](mailto:rgm@htt-consult.com)

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

Email: tiredy@cisco.com

DOTS WG  
Internet-Draft  
Intended status: Informational  
Expires: September 22, 2016

R. Dobbins, Ed.  
Arbor Networks  
S. Fouant  
Corero Network Security  
D. Migault  
Ericsson  
R. Moskowitz  
HTT Consulting  
N. Teague  
Verisign Inc  
L. Xia  
Huawei  
March 21, 2016

Use cases for DDoS Open Threat Signaling  
draft-ietf-dots-use-cases-01.txt

Abstract

This document delineates principal and ancillary use cases for DDoS Open Threat Signaling (DOTS), a communications protocol intended to facilitate the programmatic, coordinated mitigation of Distributed Denial of Service (DDoS) attacks via a standards-based mechanism. DOTS is purposely designed to support requests for DDoS mitigation services and status updates across inter-organizational administrative boundaries.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 22, 2016.



## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology and Acronyms . . . . .	4
2.1. Requirements Terminology . . . . .	4
2.2. Acronyms . . . . .	4
3. Use Cases . . . . .	4
3.1. Primary Use Cases . . . . .	6
3.1.1. Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services . . . . .	6
3.1.2. Automatic or Operator-Assisted CPE or PE Network Infrastructure Element Request to Upstream Mitigator . . . . .	8
3.1.3. Automatic or Operator-Assisted CPE or PE Attack Telemetry Detection/Classification System Request to Upstream Mitigator . . . . .	10
3.1.4. Automatic or Operator-Assisted Targeted Service/ Application Request to Upstream Mitigator . . . . .	11
3.1.5. Manual Web Portal Request to Upstream Mitigator . . . . .	13
3.1.6. Manual Mobile Device Application Request to Upstream Mitigator . . . . .	15
3.1.7. Unsuccessful Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services . . . . .	17
3.2. Ancillary Use Cases . . . . .	18
3.2.1. Auto-registration of DOTS clients with DOTS servers . . . . .	18
3.2.2. Auto-provisioning of DDoS countermeasures . . . . .	18
3.2.3. Informational DDoS attack notification to interested and authorized third parties . . . . .	19
4. Security Considerations . . . . .	19
5. IANA Considerations . . . . .	19
6. Acknowledgments . . . . .	19
7. References . . . . .	20
7.1. Normative References . . . . .	20

7.2. Informative References . . . . .	20
Authors' Addresses . . . . .	20

## 1. Introduction

Currently, distributed denial-of-service (DDoS) attack mitigation solutions/services are largely based upon siloed, proprietary communications paradigms which result in vendor/service lock-in, and as a side-effect make the configuration, provisioning, operation, and activation of these solutions a highly manual and often time-consuming process. Additionally, coordination of multiple DDoS mitigation solutions/services simultaneously engaged in defending the same organization against DDoS attacks is fraught with both technical and process-related hurdles which greatly increase operational complexity and often result in suboptimal DDoS attack mitigation efficacy.

The DDoS Open Threat Signaling (DOTS) effort is intended to facilitate interoperability between DDoS solutions/services by providing a standards-based, programmatic communications mechanism for the invitation and termination of heterogeneous DDoS attack mitigation systems and services. This allows for a much higher degree of automation and concomitant efficacy and rapidity of DDoS attack mitigation involving multiple DDoS mitigation systems and services than is currently the norm, as well as providing additional benefits such as automatic DDoS mitigation service registration and provisioning. It should be noted that DOTS is not in and of itself intended to perform orchestration functions duplicative of the functionality being developed by the [I2NSF] WG; rather, DOTS is intended to allow devices, services, and applications to request mitigation assistance and receive mitigation status updates from systems of this nature.

This document provides an overview of common DDoS mitigation system/service deployment and operational models which are in use today, but which are currently limited in scope to a single vendor or service provider and are often highly manual in nature, which can lead to miscommunications, misconfigurations, and delays in bringing mitigation services to bear against an attack. The introduction of DOTS into these scenarios will reduce reaction times and the risks associated with manual processes, simplify the use of multiple types of DDoS mitigation systems and services as required, and make practical the simultaneous use multiple DDoS mitigation systems and services as circumstances warrant.

## 2. Terminology and Acronyms

### 2.1. Requirements Terminology

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

### 2.2. Acronyms

This document makes use of the same terminology and definitions as [I-D.ietf-dots-requirements], except where noted.

## 3. Use Cases

This section provides a high-level overview of likely use cases and deployment scenarios for DOTS-enabled DDoS mitigation services. It should be noted that DOTS servers may be standalone entities which, upon receiving a DOTS mitigation service request from a DOTS client, proceed to initiate DDoS mitigation service by communicating directly or indirectly with DDoS mitigators, and likewise terminate the service upon receipt of a DOTS service termination request; conversely, the DDoS mitigators themselves may incorporate DOTS servers and/or DOTS clients. The mechanisms by which DOTS servers initiate and terminate DDoS mitigation service with DDoS mitigators is beyond the scope of this document.

All of the primary use cases described in this section are derived from current, real-world DDoS mitigation functionality, capabilities, and operational models.

The posited ancillary use cases described in this section are reasonable and highly desirable extrapolations of the functionality of baseline DOTS capabilities, and are readily attainable in the near term.

Each of the primary and ancillary use cases described in this section may be read as involving one or more DDoS mitigation service providers; DOTS makes multi-provider coordinated DDoS defenses much more effective and practical due to abstraction of the particulars of a given DDoS mitigation service/solution set.

Both the primary and ancillary use cases may be facilitated by direct DOTS client - DOTS server communications or via DOTS relays deployed in order to aggregate DOTS mitigation service requests/responses, to mediate between stateless and stateful underlying transport protocols, to aggregate multiple DOTS requests and/or responses, to

filter DOTS requests and/or responses via configured policy mechanisms, or some combination of these functions.

All DOTS messages exchanged between the DOTS clients and DOTS servers in these use cases may be communicated directly between DOTS clients and servers, or mediated by one or more DOTS relays residing on the network of the originating network, the network where upstream DDoS mitigation service takes place, an intervening network or networks, or some combination of the above.

DOTS is intended to apply to both inter- and intra-domain DDoS attack mitigation scenarios. The technical and operational requirements for inter- and intra-domain DOTS communications are identical. The main difference is administrative in nature; although it should be noted that provisioning challenges which are typically associated with inter- domain DOTS communications relationships may also apply in intra- domain deployment scenarios, based upon organizational factors. All of the same complexities surrounding authentication and authorization can apply in both contexts, including considerations such as network access policies to allow DOTS communications, DOTS transport selection (including considerations of the implications of link congestion if a stateful DOTS transport option is selected), etc. Registration of well-known ports for DOTS transports per [RFC6335] should be considered in light of these challenges.

It should also be noted that DOTS does not directly ameliorate the various administrative challenges required for successful DDoS attack mitigation. Letters of authorization, RADB updates, DNS zone delegations, alteration of network access policies, technical configurations required to facilitate network traffic diversion and re-injection, etc., are all outside the scope of DOTS. DOTS may, however, prove useful in automating the registration of DOTS clients with DOTS servers, as well as in the automatic provisioning of situationally- appropriate DDoS defenses and countermeasures. This ancillary DOTS functionality is described in Section 3.2.

Many of the 'external' administrative challenges associated with establishing workable DDoS attack mitigation service may be addressed by work currently in progress in the I2RS and I2NSF WGs. Interested parties may wish to consider tracking those efforts, and coordination with both I2RS and I2NSF is highly desirable.

Note that all the use-cases in this document are universal in nature. They apply equally to endpoint networks, transit backbone providers, cloud providers, broadband access providers, ASPs, CDNs, etc. They are not specific to particular business models, topological models, or application types, and are deliberately generalizable. Both networks targeted for attack as well as any adjacent or topologically

distant networks involved in a given scenario may be either single- or multi-homed. In the accompanying vector illustrations incorporated into draft-ietf-dots-use-cases-01.pdf, specific business and topological models are described in order to provide context.

Likewise, both DOTS itself and the use cases described in this document are completely independent of technologies utilized for the detection, classification, traceback, and mitigation of DDoS attacks. Flow telemetry such as NetFlow and IPFIX, direct full-packet analysis, log-file analysis, indirection manual observation, etc. can and will be enablers for detection, classification and traceback. Intelligent DDoS mitigation systems (IDMSes), flowspec, S/RTBH, ACLs, and other network traffic manipulation tools and techniques may be used for DDoS attack mitigation. BGP, flowspec, DNS, inline deployment, and various 'NFV' technologies may be used for network traffic diversion into mitigation centers or devices in applicable scenarios; GRE, MPLS, 'NFV', inline deployment and other techniques may be utilized for 'cleaned' traffic re-injection to its intended destination.

The scope, format, and content of all DOTS message types cited in this document must be codified by the DOTS WG.

The following use cases are intended to inform the DOTS requirements described in [I-D.ietf-dots-requirements].

### 3.1. Primary Use Cases

#### 3.1.1. Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services

One or more CPE or PE mitigators with DOTS client capabilities may be configured to signal to one or more DOTS servers in order to request upstream DDoS mitigation service initiation during an attack when DDoS attack volumes and/or attack characteristics exceed the capabilities of such CPE mitigators. DDoS mitigation service may be terminated either automatically or manually via a DOTS mitigation service termination request initiated by the mitigator when it has been determined that the DDoS attack has ended.

- (a) A DDoS attack is initiated against online properties of an organization which has deployed DOTS-client-capable DDoS mitigators.
- (b) CPE or PE DDoS mitigators detect, classify, and begin mitigating the DDoS attack.

- (c) CPE or PE DDoS mitigators determine that their capacity and/or capability to mitigate the DDoS attack is insufficient, and utilize their DOTS client functionality to send a DOTS mitigation service initiation request to one or more DOTS servers residing on one or more upstream transit networks, peer networks, or overlay MSSP networks. This DOTS mitigation service initiation request may be automatically initiated by the CPE or PE DDoS mitigators, or may be manually triggered by personnel of the requesting organization in response to an alert from the mitigators (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been configured to honor requests from the requesting CPE or PE mitigators, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (e) The DOTS servers transmit a DOTS service status message to the requesting CPE or PE mitigators indicating that upstream DDoS mitigation service has been initiated.
- (f) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the requesting CPE or PE mitigators.
- (g) While DDoS mitigation services are active, the CPE or PE mitigators may optionally regularly transmit DOTS mitigation efficacy updates to the relevant DOTS servers.
- (h) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (i) The DOTS servers transmit a DOTS mitigation status update to the CPE or PE mitigators indicating that the DDoS attack has ceased.
- (j) The CPE or PE DDoS mitigators transmit a DOTS mitigation service termination request to the DOTS servers. This DOTS mitigation service termination request may be automatically initiated by the CPE or PE DDoS mitigators, or may be manually triggered by personnel of the requesting organization in response to an alert from the mitigators or a management system which monitors them (the mechanism by which this process takes place is beyond the scope of this document).

- (k) The DOTS servers terminate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (l) The DOTS servers transmit a DOTS mitigation status update to the CPE or PE mitigators indicating that DDoS mitigation services have been terminated.
- (m) The CPE or PE DDoS mitigators transmit a DOTS mitigation termination status acknowledgement to the DOTS servers.

### 3.1.2. Automatic or Operator-Assisted CPE or PE Network Infrastructure Element Request to Upstream Mitigator

CPE or PE network infrastructure elements such as routers, switches, load-balancers, firewalls, 'IPSeS', etc. which have the capability to detect and classify DDoS attacks and which have DOTS client capabilities may be configured to signal to one or more DOTS servers in order to request upstream DDoS mitigation service initiation during an attack. DDoS mitigation service may be terminated either automatically or manually via a DOTS mitigation service termination request initiated by the network element when it has been determined that the DDoS attack has ended.

In this use-case, the network elements involved are not engaged in mitigating DDoS attack traffic. They are signaling for upstream attack mitigation assistance. This can be an inter- or intra- domain use-case.

- (a) A DDoS attack is initiated against online properties of an organization with DOTS-client-capable network infrastructure elements deployed.
- (b) The network infrastructure elements utilize their DOTS client functionality to send a DOTS mitigation service initiation request to one or more DOTS servers residing on one or more upstream transit networks, peer networks, or overlay MSSP networks, either directly or via intermediate DOTS relays residing upon the requesting organization's network, the upstream mitigation provider's network, or both. The scope, format, and content of these messages must be codified by the DOTS WG. This DOTS mitigation service initiation request may be automatically initiated by the network infrastructure elements, or may be manually triggered by personnel of the requesting organization in response to an alert from the network elements or a management system which monitors them (the mechanism by which this process takes place is beyond the scope of this document).

- (c) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been configured to honor requests from the requesting network infrastructure elements, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers transmit a DOTS service status message to the requesting network infrastructure elements indicating that upstream DDoS mitigation service has been initiated.
- (e) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the requesting requesting network infrastructure elements.
- (f) While DDoS mitigation services are active, the network infrastructure elements may optionally regularly transmit DOTS mitigation efficacy updates to the relevant DOTS servers.
- (g) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (h) The DOTS servers transmit a DOTS mitigation status update to the network infrastructure elements indicating that the DDoS attack has ceased.
- (i) The network infrastructure elements transmit a DOTS mitigation service termination request to the DOTS servers. This DOTS mitigation service termination request may be automatically initiated by the network infrastructure elements, or may be manually triggered by personnel of the requesting organization in response to an alert from the mitigators (the mechanism by which this process takes place is beyond the scope of this document).
- (j) The DOTS servers terminate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (k) The DOTS servers transmit a DOTS mitigation status update to the network infrastructure elements indicating that DDoS mitigation services have been terminated.
- (l) The network infrastructure elements transmit a DOTS mitigation termination status acknowledgement to the DOTS servers.



### 3.1.3. Automatic or Operator-Assisted CPE or PE Attack Telemetry Detection/Classification System Request to Upstream Mitigator

CPE or PE Attack Telemetry Detection/Classification Systems which have DOTS client capabilities may be configured so that upon detecting and classifying a DDoS attack, they signal one or more DOTS servers in order to request upstream DDoS mitigation service initiation. DDoS mitigation service may be terminated either automatically or manually via a DOTS mitigation service termination request initiated by the Attack Telemetry Detection/Classification System when it has been determined that the DDoS attack has ended.

In this use-case, the Attack Telemetry Detection/Classification does not possess any inherent capability to mitigate DDoS attack traffic, and is signaling for upstream mitigation assistance. This can be an inter- or intra-domain use-case.

- (a) A DDoS attack is initiated against online properties of an organization with DOTS-client-capable CPE or PE Attack Telemetry Detection/Classification Systems deployed.
- (b) The CPE or PE Attack Telemetry Detection/Classification Systems utilize their DOTS client functionality to send a DOTS mitigation service initiation request to one or more DOTS servers residing on one or more upstream transit networks, peer networks, or overlay MSSP networks, either directly or via intermediate DOTS relays residing upon the requesting organization's network, the upstream mitigation provider's network, or both. This DOTS mitigation service initiation request may be automatically initiated by the CPE or PE Attack Telemetry Detection/Classification Systems, or may be manually triggered by personnel of the requesting organization in response to an alert from the CPE or PE Attack Telemetry Detection/Classification Systems (the mechanism by which this process takes place is beyond the scope of this document).
- (c) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been configured to honor requests from the requesting CPE or PE Attack Telemetry Detection/Classification Systems, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers transmit a DOTS service status message to the requesting CPE or PE Attack Telemetry Detection/Classification Systems indicating that upstream DDoS mitigation service has been initiated.

- (e) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the requesting CPE or PE Attack Telemetry Detection/Classification Systems.
- (f) While DDoS mitigation services are active, the CPE or PE Attack Telemetry Detection/Classification Systems may optionally regularly transmit DOTS mitigation efficacy updates to the relevant DOTS servers.
- (g) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (h) The DOTS servers transmit a DOTS mitigation status update to the CPE or PE Attack Telemetry Detection/Classification Systems indicating that the DDoS attack has ceased.
- (i) The CPE or PE Attack Telemetry Detection/Classification Systems transmit a DOTS mitigation service termination request to the DOTS servers. This DOTS mitigation service termination request may be automatically initiated by the CPE or PE Attack Telemetry Detection/Classification Systems, or may be manually triggered by personnel of the requesting organization in response to an alert from the CPE or PE Attack Telemetry Detection/Classification Systems (the mechanism by which this process takes place is beyond the scope of this document).
- (j) The DOTS servers terminate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (k) The DOTS servers transmit a DOTS mitigation status update to the CPE or PE Attack Telemetry Detection/Classification Systems indicating that DDoS mitigation services have been terminated.
- (l) The CPE or PE Attack Telemetry Detection/Classification Systems transmit a DOTS mitigation termination status acknowledgement to the DOTS servers.

#### 3.1.4. Automatic or Operator-Assisted Targeted Service/ Application Request to Upstream Mitigator

A service or application which is the target of a DDoS attack and which has the capability to detect and classify DDoS attacks (i.e., Apache mod\_security [APACHE], BIND RRL [RRL], etc.) as well as DOTS client functionality may be configured so that upon detecting and

classifying a DDoS attack, it signals one or more DOTS servers in order to request upstream DDoS mitigation service initiation. DDoS mitigation service may be terminated either automatically or manually via a DOTS mitigation service termination request initiated by the service/application when it has been determined that the DDoS attack has ended.

In this use-case, the service/application does not possess inherent DDoS attack mitigation capabilities, and is signaling for upstream mitigation assistance. This can be an inter- or intra-domain use-case.

- (a) A DDoS attack is initiated against online properties of an organization which include DOTS-client-capable services or applications that are the specific target(s) of the attack.
- (b) The targeted services or applications utilize their DOTS client functionality to send a DOTS mitigation service initiation request to one or more DOTS servers residing on the same network as the services or applications, one or more upstream transit networks, peer networks, or overlay MSSP networks, either directly or via intermediate DOTS relays residing upon the requesting organization's network, the upstream mitigation provider's network, or both. This DOTS mitigation service initiation request may be automatically initiated by the targeted services or applications, or may be manually triggered by personnel of the requesting organization in response to an alert from the targeted services or applications or a system which monitors them (the mechanism by which this process takes place is beyond the scope of this document).
- (c) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been provisioned to honor requests from the requesting services or applications, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers transmit a DOTS service status message to the services or applications indicating that upstream DDoS mitigation service has been initiated
- (e) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the requesting services or applications.

- (f) While DDoS mitigation services are active, the requesting services or applications may optionally regularly transmit DOTS mitigation efficacy updates to the relevant DOTS servers.
- (g) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (h) The DOTS servers transmit a DOTS mitigation status update to the requesting services or applications indicating that the DDoS attack has ceased.
- (i) The targeted services or applications transmit a DOTS mitigation service termination request to the DOTS servers. This DOTS mitigation service termination request may be automatically initiated by the targeted services or applications, or may be manually triggered by personnel of the requesting organization in response to an alert from a system which monitors them (the mechanism by which this process takes place is beyond the scope of this document).
- (j) The DOTS servers terminate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (k) The DOTS servers transmit a DOTS mitigation status update to the targeted services or applications indicating that DDoS mitigation services have been terminated.
- (l) The targeted services or applications transmit a DOTS mitigation termination status acknowledgement to the DOTS servers.

#### 3.1.5. Manual Web Portal Request to Upstream Mitigator

A Web portal which has DOTS client capabilities has been configured in order to allow authorized personnel of organizations which are targeted by DDoS attacks to manually request upstream DDoS mitigation service initiation from a DOTS server. When an organization has reason to believe that it is under active attack, authorized personnel may utilize the Web portal to manually initiate a DOTS client mitigation request to one or more DOTS servers. DDoS mitigation service may be terminated manually via a DOTS mitigation service termination request through the Web portal when it has been determined that the DDoS attack has ended.

In this use-case, the organization targeted for attack does not possess any automated or operator-assisted mechanisms for DDoS attack

detection, classification, traceback, or mitigation; the existence of an attack has been inferred manually, and the organization is requesting upstream mitigation assistance. This can theoretically be an inter- or intra-domain use-case, but is more typically an inter-domain scenario.

- (a) A DDoS attack is initiated against online properties of an organization have access to a Web portal which incorporates DOTS client functionality and can generate DOTS mitigation service requests upon demand.
- (b) Authorized personnel utilize the Web portal to send a DOTS mitigation service initiation request to one or more upstream transit networks, peer networks, or overlay MSSP networks, either directly or via intermediate DOTS relays residing upon the requesting organization's network, the upstream mitigation provider's network, or both. This DOTS mitigation service initiation request is manually triggered by personnel of the requesting organization when it is judged that the organization is under DDoS attack (the mechanism by which this process takes place is beyond the scope of this document).
- (c) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been provisioned to honor requests from the Web portal, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers transmit a DOTS service status message to the Web portal indicating that upstream DDoS mitigation service has been initiated.
- (e) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the Web portal.
- (f) While DDoS mitigation services are active, the Web portal may optionally regularly transmit manually-triggered DOTS mitigation efficacy updates to the relevant DOTS servers.
- (g) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (h) The DOTS servers transmit a DOTS mitigation status update to the Web portal indicating that the DDoS attack has ceased.

- (i) The Web portal transmits a manually-triggered DOTS mitigation service termination request to the DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (j) The Web portal transmits a manually-triggered DOTS mitigation service termination request to the DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (k) The DOTS servers transmit a DOTS mitigation status update to the Web portal indicating that DDoS mitigation services have been terminated.
- (l) The Web portal transmits a DOTS mitigation termination status acknowledgement to the DOTS servers.

#### 3.1.6. Manual Mobile Device Application Request to Upstream Mitigator

An application for mobile devices such as smartphones and tablets which incorporates DOTS client capabilities has been made available to authorized personnel of an organization. When the organization has reason to believe that it is under active DDoS attack, authorized personnel may utilize the mobile device application to manually initiate a DOTS client mitigation request to one or more DOTS servers in order to initiate upstream DDoS mitigation services. DDoS mitigation service may be terminated manually via a DOTS mitigation service termination request initiated through the mobile device application when it has been determined that the DDoS attack has ended.

This use-case is similar to the one described in Section 3.1.5; the difference is that a mobile application provided by the DDoS mitigation service provider is used to request upstream attack mitigation assistance. This can theoretically be an inter- or intra-domain use-case, but is more typically an inter-domain scenario.

- (a) A DDoS attack is initiated against online properties of an organization have access to a Web portal which incorporates DOTS client functionality and can generate DOTS mitigation service requests upon demand.
- (b) Authorized personnel utilize the mobile application to send a DOTS mitigation service initiation request to one or more DOTS servers residing on the same network as the targeted Internet properties, one or more upstream transit networks, peer networks, or overlay MSSP networks, either directly or via intermediate DOTS relays residing upon the requesting

organization's network, the upstream mitigation provider's network, or both. This DOTS mitigation service initiation request is manually triggered by personnel of the requesting organization when it is judged that the organization is under DDoS attack (the mechanism by which this process takes place is beyond the scope of this document).

- (c) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been provisioned to honor requests from the mobile application, and initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers transmit a DOTS service status message to the mobile application indicating that upstream DDoS mitigation service has been initiated.
- (e) While DDoS mitigation services are active, the DOTS servers regularly transmit DOTS mitigation status updates to the mobile application.
- (f) While DDoS mitigation services are active, the mobile application may optionally regularly transmit manually-triggered DOTS mitigation efficacy updates to the relevant DOTS servers.
- (g) When the upstream DDoS mitigators determine that the DDoS attack has ceased, they indicate this change in status to their respective DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (h) The DOTS servers transmit a DOTS mitigation status update to the mobile application indicating that the DDoS attack has ceased.
- (i) The mobile application transmits a manually-triggered DOTS mitigation service termination request to the DOTS servers (the mechanism by which this process takes place is beyond the scope of this document).
- (j) The DOTS servers terminate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (k) The DOTS servers transmit a DOTS mitigation status update to the mobile application indicating that DDoS mitigation services have been terminated.

- (1) The mobile application transmits a DOTS mitigation termination status acknowledgement to the DOTS servers.

#### 3.1.7. Unsuccessful Automatic or Operator-Assisted CPE or PE Mitigators Request Upstream DDoS Mitigation Services

One or more CPE or PE mitigators with DOTS client capabilities may be configured to signal to one or more DOTS servers in order to request upstream DDoS mitigation service initiation during an attack when DDoS attack volumes and/or attack characteristics exceed the capabilities of such CPE mitigators. DDoS mitigation service may be terminated either automatically or manually via a DOTS mitigation service termination request initiated by the mitigator when it has been determined that the DDoS attack has ended.

This can theoretically be an inter- or intra-domain use-case, but is more typically an inter-domain scenario.

- (a) A DDoS attack is initiated against online properties of an organization which has deployed DOTS-client-capable DDoS mitigators.
- (b) CPE or PE DDoS mitigators detect, classify, and begin mitigating the DDoS attack.
- (c) CPE or PE DDoS mitigators determine that their capacity and/or capability to mitigate the DDoS attack is insufficient, and utilize their DOTS client functionality to send a DOTS mitigation service initiation request to one or more DOTS servers residing on one or more upstream transit networks, peer networks, or overlay MSSP networks. This DOTS mitigation service initiation request may be automatically initiated by the CPE or PE DDoS mitigators, or may be manually triggered by personnel of the requesting organization in response to an alert from the mitigators (the mechanism by which this process takes place is beyond the scope of this document).
- (d) The DOTS servers which receive the DOTS mitigation service initiation requests determine that they have been configured to honor requests from the requesting CPE or PE mitigators, and attempt to initiate situationally-appropriate DDoS mitigation service on their respective networks (the mechanism by which this process takes place is beyond the scope of this document).
- (e) The DDoS mitigators on the upstream network report back to the DOTS servers that they are unable to initiate DDoS mitigation service for the requesting organization due to mitigation capacity constraints, bandwidth constraints, functionality



constraints, hardware casualties, or other impediments (the mechanism by which this process takes place is beyond the scope of this document).

- (f) The DOTS servers transmit a DOTS service status message to the requesting CPE or PE mitigators indicating that upstream DDoS mitigation service cannot be initiated as requested.
- (g) The CPE or PE mitigators may optionally regularly re-transmit DOTS mitigation status request messages to the relevant DOTS servers until acknowledgement that mitigation services have been initiated.
- (h) The CPE or PE mitigators may optionally transmit a DOTS mitigation service initiation request to DOTS servers associated with a configured fallback upstream DDoS mitigation service. Multiple fallback DDoS mitigation services may optionally be configured.
- (i) The process describe above cyclically continues until the DDoS mitigation service request is fulfilled; the CPE or PE mitigators determine that the DDoS attack volume has decreased to a level and/or complexity which they themselves can successfully mitigate; the DDoS attack has ceased; or manual intervention by personnel of the requesting organization has taken place.

### 3.2. Ancillary Use Cases

#### 3.2.1. Auto-registration of DOTS clients with DOTS servers

An additional benefit of DOTS is that by utilizing agreed-upon authentication mechanisms, DOTS clients can automatically register for DDoS mitigation service with one or more upstream DOTS servers. The details of such registration are beyond the scope of this document.

#### 3.2.2. Auto-provisioning of DDoS countermeasures

The largely manual tasks associated with provisioning effective, situationally-appropriate DDoS countermeasures is a significant barrier to providing/obtaining DDoS mitigation services for both mitigation providers and mitigation recipients. Due to the 'self-descriptive' nature of DOTS registration messages and mitigation requests, the implementation and deployment of DOTS has the potential to automate countermeasure selection and configuration for DDoS mitigators. The details of such provisioning are beyond the scope of this document.

This can theoretically be an inter- or intra-domain use-case, but is more typically an inter-domain scenario.

### 3.2.3. Informational DDoS attack notification to interested and authorized third parties

In addition to its primary role of providing a standardized, programmatic approach to the automated and/or operator-assisted request of DDoS mitigation services and providing status updates of those mitigations to requesters, DOTS may be utilized to notify security researchers, law enforcement agencies, regulatory bodies, etc. of DDoS attacks against attack targets, assuming that organizations making use of DOTS choose to share such third-party notifications, in keeping with all applicable laws, regulations, privacy and confidentiality considerations, and contractual agreements between DOTS users and said third parties.

This is an inter-domain scenario.

## 4. Security Considerations

DOTS is at risk from three primary attacks: DOTS agent impersonation, traffic injection, and signaling blocking. The DOTS protocol MUST be designed for minimal data transfer to address the blocking risk.

Impersonation and traffic injection mitigation can be managed through current secure communications best practices. DOTS is not subject to anything new in this area. One consideration could be to minimize the security technologies in use at any one time. The more needed, the greater the risk of failures coming from assumptions on one technology providing protection that it does not in the presence of another technology.

Additional details of DOTS security requirements may be found in [I-D.ietf-dots-requirements].

## 5. IANA Considerations

No IANA considerations exist for this document at this time.

## 6. Acknowledgments

TBD

## 7. References

### 7.1. 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>>.

### 7.2. Informative References

- [APACHE] "Apache mod\_security", <<https://www.modsecurity.org>>.
- [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.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<http://www.rfc-editor.org/info/rfc6335>>.
- [RRL] "BIND RRL", <<https://deepthought.isc.org/article/AA-00994/0/Using-the-Response-Rate-Limiting-Feature-in-BIND-9.10.html>>.

### Authors' Addresses

Roland Dobbins (editor)  
Arbor Networks  
30 Raffles Place  
Level 17 Chevron House  
Singapore 048622  
Singapore

Email: [rdobbins@arbor.net](mailto:rdobbins@arbor.net)

Stefan Fouant  
Corero Network Security

Email: [Stefan.Fouant@corero.com](mailto:Stefan.Fouant@corero.com)

Daniel Migault  
Ericsson  
8400 boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Phone: +1 514-452-2160  
Email: daniel.migault@ericsson.com

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 48237  
USA

Email: rgm@labs.htt-consult.com

Nik Teague  
Verisign Inc  
12061 Bluemont Way  
Reston, VA 20190  
USA

Phone: +44 791 763 5384  
Email: nteague@verisign.com

Liang Xia  
Huawei  
No. 101, Software Avenue, Yuhuatai District  
Nanjing  
China

Email: Frank.xialiang@huawei.com

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

A. Mortensen  
Arbor Networks, Inc.  
F. Andreassen  
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

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

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

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

This Internet-Draft will expire on September 20, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Context and Motivation . . . . .	2
1.1. Terminology . . . . .	3
1.1.1. Key Words . . . . .	3
1.1.2. Definition of Terms . . . . .	3
1.2. Scope . . . . .	3
1.3. Assumptions . . . . .	4
2. Architecture . . . . .	5
2.1. DOTS Operations . . . . .	7
2.2. DOTS Agent Relationships . . . . .	9
3. Components . . . . .	13
3.1. DOTS client . . . . .	13
3.2. DOTS server . . . . .	14
4. Concepts . . . . .	15
4.1. Signaling Sessions . . . . .	15
4.1.1. Preconditions . . . . .	15
4.1.2. Establishing the Signaling Session . . . . .	15
4.1.3. Maintaining the Signaling Session . . . . .	16
4.2. Modes of Signaling . . . . .	17
4.2.1. Direct Signaling . . . . .	17
4.2.2. Relayed Signaling . . . . .	17
4.2.3. Redirected Signaling . . . . .	18
4.2.4. Recursive Signaling . . . . .	19
5. Security Considerations . . . . .	22
6. Acknowledgments . . . . .	23
7. Change Log . . . . .	23
8. References . . . . .	23
8.1. Normative References . . . . .	23
8.2. Informative References . . . . .	23
Authors' Addresses . . . . .	24

## 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:

- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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:

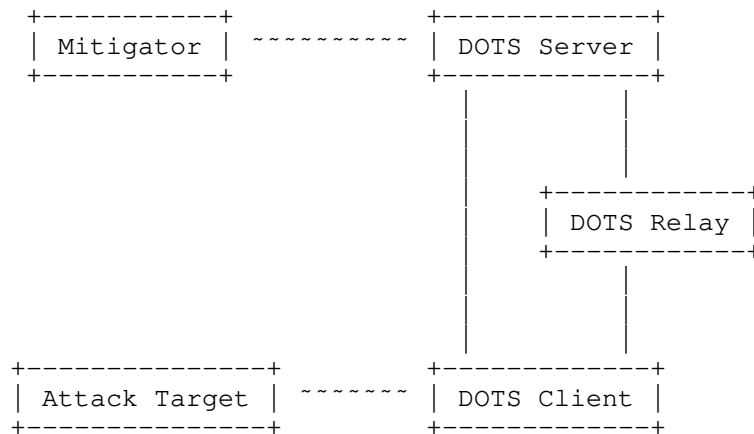


Figure 1: Basic DOTS Architecture

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):

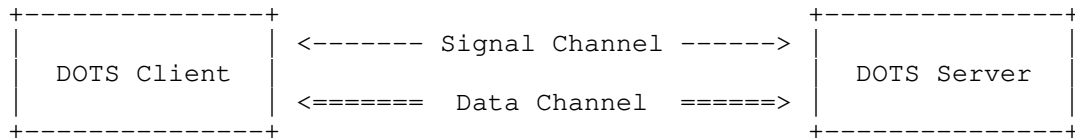


Figure 2: DOTS Interfaces

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 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 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):

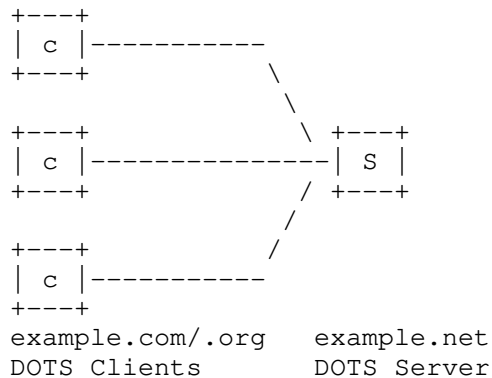


Figure 3: Multiple DOTS clients for a DOTS server

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.]]

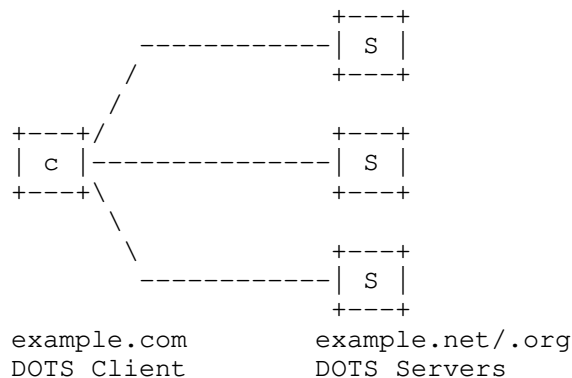


Figure 4: Multi-Homed DOTS Client

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.

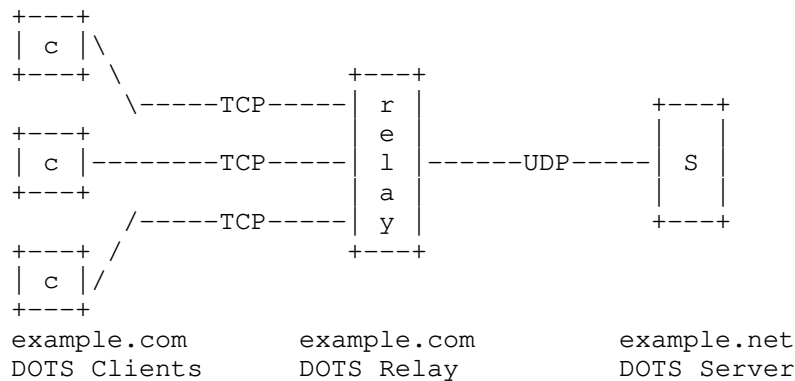


Figure 5: Client-Side Relay with Aggregation

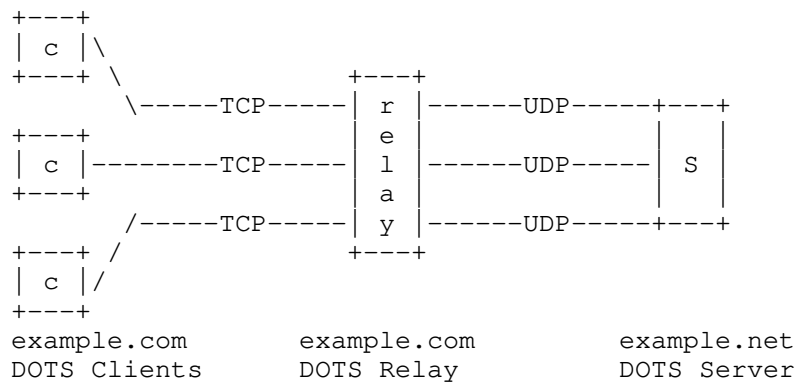


Figure 6: Client-Side Relay without Aggregation

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.]]

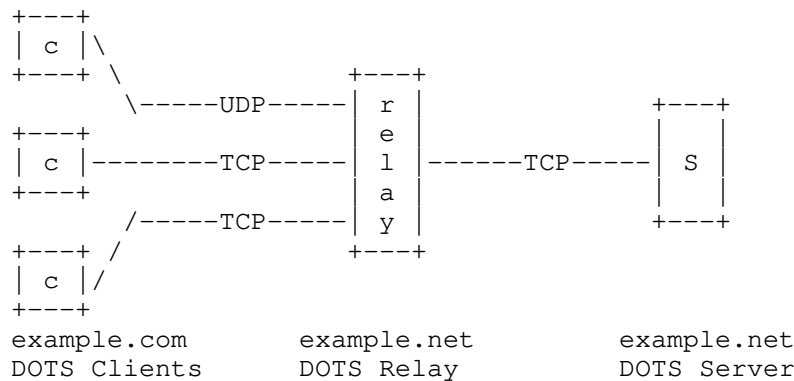


Figure 7: Server-Side Relay with Aggregation

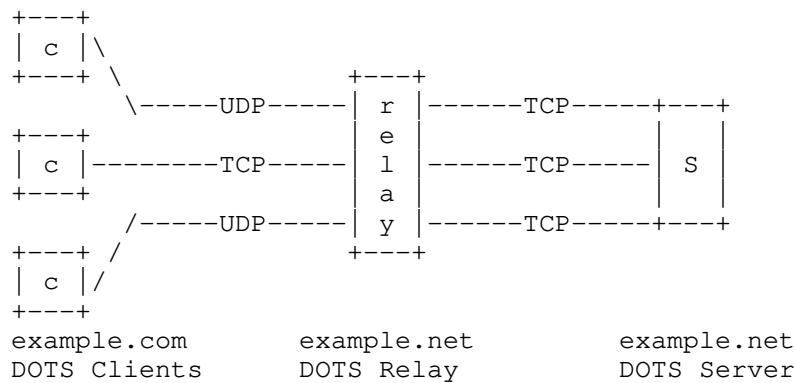


Figure 8: Server-Side Relay without Aggregation

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:

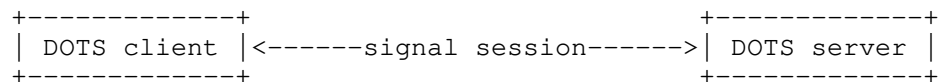


Figure 9: Direct Signaling

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:

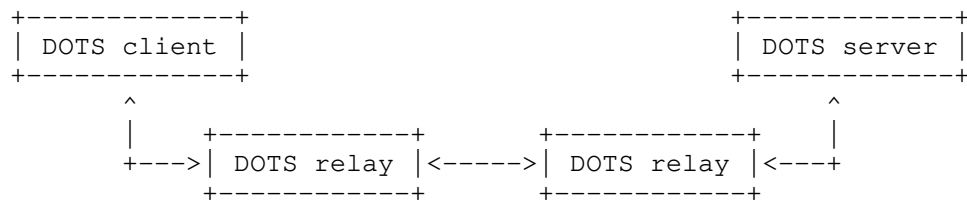


Figure 10: Relayed Signaling

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:

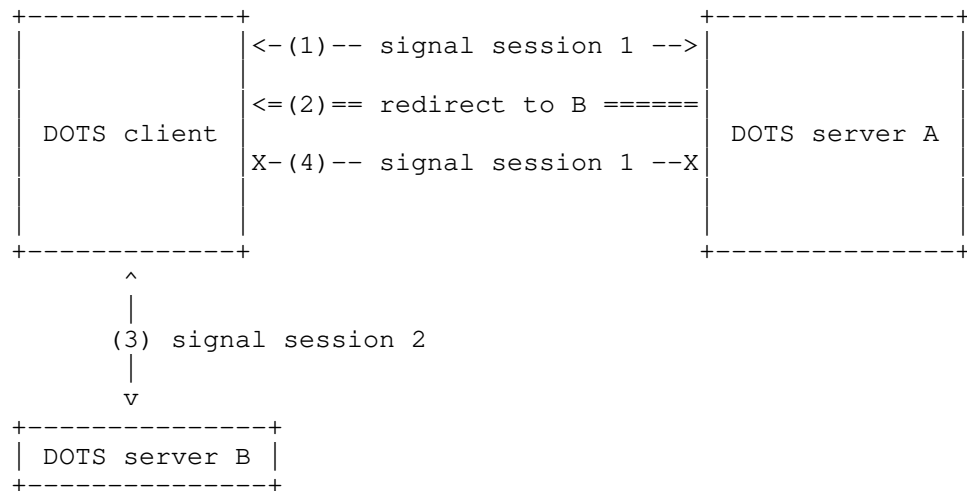


Figure 11: Redirected Signaling

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:



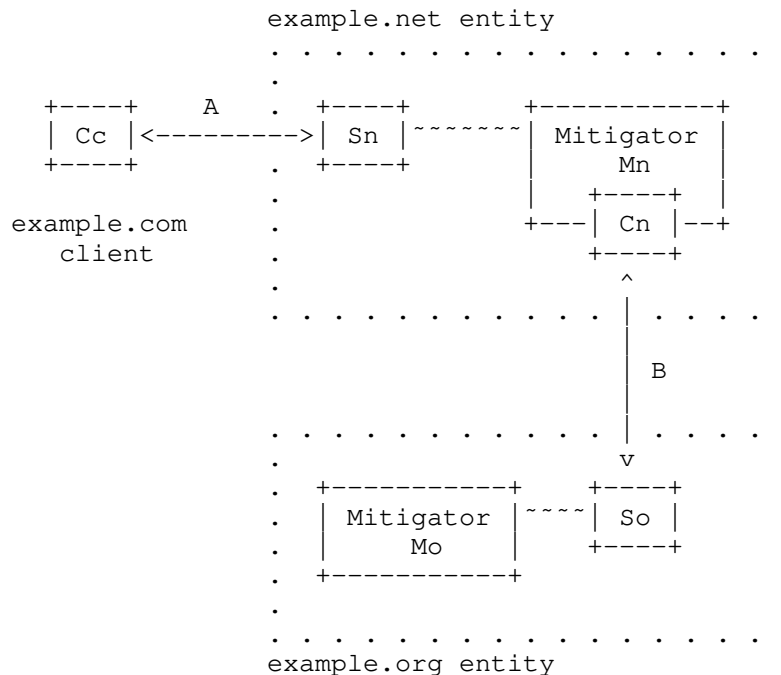


Figure 12: Recursive Signaling

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](mailto:amortensen@arbor.net)

Flemming Andreassen  
Cisco Systems, Inc.  
United States

EMail: [fandreas@cisco.com](mailto:fandreas@cisco.com)

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

EMail: [tiredy@cisco.com](mailto:tiredy@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

DOTS  
Internet-Draft  
Intended status: Standards Track  
Expires: August 6, 2016

R. Moskowitz  
J. Xia  
Huawei  
February 3, 2016

DOTS over GRE  
draft-moskowitz-dots-gre-00.txt

Abstract

This document describes using a GRE tunnel to deliver DOTS messages between DOTS agents and compares it to other methods.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 6, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terms and Definitions . . . . .	2
2.1. Requirements Terminology . . . . .	2
3. Problem Space . . . . .	3
3.1. The Network issues faced by DOTS and UDP . . . . .	3
3.2. Peer-to-peer not RESTful . . . . .	3
3.3. Security Context . . . . .	3
3.3.1. Stateful Security Context . . . . .	3
3.3.2. Security Context and Fate Sharing . . . . .	3
4. Protocol Selection Considerations . . . . .	4
5. The DOTS Protocol Stack . . . . .	5
5.1. GRE full stack tunnel . . . . .	5
5.1.1. Design Analysis . . . . .	5
5.2. GRE with compressed stack tunnel . . . . .	5
5.3. ESP transport mode . . . . .	6
6. Management Considerations . . . . .	6
6.1. DOTS agent connectivity management . . . . .	6
6.2. Secure Context management . . . . .	6
7. IANA Considerations . . . . .	7
8. Security Considerations . . . . .	7
9. Contributors . . . . .	8
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

This document describes using a GRE [RFC2784] tunnel to deliver DOTS messages between DOTS agents. Various alternatives for transporting DOTS messages are analyzed and the justification of GRE over alternatives as UDP over IP and UDP over ESP over IP is presented.

The intent of this document is to encourage discussion on the most effective set of protocols to provide the high reliability requirement spelled out in the DOTS requirements document [I-D.ietf-dots-requirements].

## 2. Terms and Definitions

## 2.1. Requirements Terminology

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

### 3. Problem Space

#### 3.1. The Network issues faced by DOTS and UDP

DOTS messaging needs to occur during the worst time to expect reliable packet delivery. That is during a DDoS attack. Not only is link to (and potentially from) the DOTS client fully congested with the attack, but also the ISPs between the attacked DOTS client and the responsible DOTS server may have instituted UDP blocking mitigation activities.

It is this UDP blocking mitigation action that presents a double-edged effect. It lessens the impact of the attack, allowing TCP-based activity to continue. It stops any attack management, i.e. DOTS, messaging over UDP to traverse the portion of the network where the blocking is in affect.

#### 3.2. Peer-to-peer not RESTful

A second problem, or more a challenge, in DOTS messaging is that it is really a peer communication. That is the DOTS server may be messaging the DOTS client at any time, including during an attack. Thus a client-service approach like RESTful would require 2 uni-directional sessions.

One example of a DOTS server message is an "Attack seems over" message from the server to the client.

#### 3.3. Security Context

##### 3.3.1. Stateful Security Context

Security Context is the collection of information to manage the securing of information. In this case DOTS messages. The only viable method for stateless security is secure data objects as in PEM [RFC1421]. Stateless security is very resource intensive and typically avoided unless it is the only effective approach. DOTS messaging will use a secure data channel which is stateful. This state needs to be managed and protected.

##### 3.3.2. Security Context and Fate Sharing

Security Context often contains communication protocol information like IP addresses and transport ports. In these situations the security context is said to "share fate" with these aspects of the communications. If something disrupts the communication state, it disrupts the security context, often requiring some degree of security re-initialization.



The greater the fate sharing, the more rigid the security context and more prone to attack. Thus a secure message transport design goal is to lessen the degree of fate sharing.

#### 4. Protocol Selection Considerations

Based on Section 3, DOTS messaging should take advantage of protocols that:

- o Are bi-directional

One such protocol is often used for bi-directional messaging is TCP. This is not a viable option as the ACK from sending a message from the DOTS client to server over the potentially uncongested uplink may never get back to the client over the congested down link.

- o Are Not Commonly blocked, particularly during a DDoS attack

UDP and ICMP fall into this avoidance category.

- o Have minimal overhead

DOTS messages that are sent during an attack should fit into a single MTU. The lower the protocol byte overhead, the more space available for the DOTS message itself.

- o Are only enabled by need on a system

It would be advantageous that the DOTS communication uses a protocol that is typically quickly discarded by most targeted systems. Even though these protocols are used by the DOTS agents, the DOTS agents will be hard to find to attack and will tend to have more resources available to deflect direct attacks.

- o Support peer communications

At all protocol levels, there must be no complexities in implementing peer communications. Pairing two uni-directional protocols to achieve this should be avoided.

The security context that protects the DOTS messaging must support peer communications. That is a single DOTS agent security agreement would provide the complete context for DOTS security. Examples include IKEv2 [RFC5996] and HIPv2 [RFC7401]. It is noted that these maintain two uni-directional Security Associations within the security context to properly manage the key usage in each direction.

- o Provide secure communications with minimal fate-sharing

The security context should be resilient to DOTS agent restart and thus potential loss of protocol state. At best there should be no fate-sharing with any protocol state. An option for security state to be stored in a safe manner so that it need not be renegotiated after agent restart makes forcing an agent restart an uninteresting attack.

## 5. The DOTS Protocol Stack

Below are three possible protocol designs. The compressed GRE design, Section 5.2, best meets the selection considerations (Section 4).

### 5.1. GRE full stack tunnel

GRE is basically used to tunnel Ethernet payloads across an IP network. For example an IPv4 datagram can be tunneled within GRE with a GRE Protocol Type of 0x800. This is simple to implement on a system, as GRE appears to IP as an interface. DOTS messaging can be secured with SSE [I-D.moskowitz-sse] on UDP over IPv4 or IPv6 within this GRE tunnel.

GRE can also work well in a NAT traversal deployment scenario.

#### 5.1.1. Design Analysis

The per-packet byte cost of GRE and an inner IP envelope (IPv4 or IPv6) is balanced in part by the envelope simplicity of SSE. SSE has the advantage of being completely free of fate-sharing with the lower protocol levels. GRE, as indicated, is relatively easy to support as a pseudo-interface. This is weighed against SSE being new, and any Key Management Protocol would need negotiation parameters to support SSE.

Use of SSE also allows secure transport of DOTS messages over non-IP connections, for example SMS. The low SSE envelope overhead of as little as 20 bytes can allow for 120 bytes for a single SMS message. SMS message continuation can allow for longer DOTS messages.

### 5.2. GRE with compressed stack tunnel

The full GRE stack approach may overly constrain the size of the DOTS message that can fit within a single MTU. There are approaches to compress this into a smaller size.

There are two approaches to reduce the header overhead of the GRE full stack tunnel outlined above. RObust Header Compression [RFC3095] is the well-known approach. Within this compression, the datagram will logically be the same as above.

The actual inner IP header could be compressed to zero bytes by using the same source and destination addresses of the outer IP header. This is more than specified in ROHC, and would involve additional specification. NAT traversal design considerations need to be included in the compression scheme.

### 5.3. ESP transport mode

ESP [RFC4303] in transport mode (or BEET with HIPv2) Provides a familiar approach to protect UDP traffic. ESP with IKEv2 fate-shares with both IP and UDP. ESP with HIPv2 only with UDP. Either way, loss of UDP state due to a DOTS server crash would require reestablishment of the security state. This keeps attacks against the DOTS server as an important attack surface to weigh against the familiarity of ESP with IKEv2 or HIP.

ESP limits secure DOTS messaging to IP networks. A different method would be needed for sending DOTS messages over SMS or require IP over a modem connection.

ESP NAT traversal uses UDP and thus reintroduces the UDP blocking concern discussed above.

## 6. Management Considerations

### 6.1. DOTS agent connectivity management

A DOTS client needs to be configured with knowledge of the DOTS servers. This may either by an IP address or an FQDN. If FQDN is used, IP addresses should be cached as DNS lookups may fail during an attack.

### 6.2. Secure Context management

Some trustworthy authentication needs to be set up on both sides. This authentication knowledge will be used by a Key Management Protocol like IKEv2 or HIPv2 to create the security context. Either can manage the security context for ESP or SSE. Two strong authentication methods use digital certificates or raw public keys.

Digital certificate trustworthiness may not be easy to determine. There are many issues such as which Certificate Authority to trust and how to manage Certificate Domain trust leakage. These issues

often result in needing to manage an authorization list of trusted certificates.

Raw public keys for IKEv2 [I-D.kivinen-ipsecme-oob-pubkey] or HIPv2 HITs can be managed in an ACL without the cost associated with Digital Certificates. Replacing 'old' keys can be associated with the DOTS business model of contract renewal.

## 7. IANA Considerations

No IANA considerations exist for this document at this time.

## 8. Security Considerations

A DDoS attacker would greatly benefit from disabling DOTS. This may be accomplished by:

- o Blocking DOTS traffic.
- o Disabling DOTS servers.
- o Disabling DOTS clients.

A key component of this proposal is to lessen the likelihood of ISPs from blocking DOTS traffic by not using UDP. Whatever protocol DOTS uses, may be used in future DDoS attacks, but will not be as effective as UDP based attacks. Thus not using UDP is a worthwhile goal.

DOTS server resiliency to attacks is a critical goal. Loss of a DOTS server can impact many clients (customers). The less fate-sharing the higher the attack resiliency, which is why this document recommends the GRE with compressed stack tunnel, Section 5.2, approach.

DOTS clients will tend to be invisible to attackers, but over time they will be discovered for targeted attacks, thus the same resiliency considerations applied to the servers also apply to the clients. Additionally, DOTS clients should avoid access to as many Internet services as possible, as at critical times they may be blocked. Thus a non-PKI authentication scheme as in raw public keys has the advantage of needing one less Internet resource that may be blocked.

## 9. Contributors

The following contributed actively to the this document: Sue Hares (Huawei)

## 10. References

### 10.1. 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>>.

### 10.2. 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.kivinen-ipsecme-oob-pubkey] Kivinen, T., Wouters, P., and H. Tschofenig, "Generic Raw Public Key Support for IKEv2", draft-kivinen-ipsecme-oob-pubkey-14 (work in progress), October 2015.
- [I-D.moskowitz-sse] Moskowitz, R., Faynberg, I., <>, H., Hares, S., and P. Giacomini, "Session Security Envelope", draft-moskowitz-sse-01 (work in progress), January 2016.
- [RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <<http://www.rfc-editor.org/info/rfc1421>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<http://www.rfc-editor.org/info/rfc2784>>.
- [RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <<http://www.rfc-editor.org/info/rfc3095>>.

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, DOI 10.17487/RFC5996, September 2010, <<http://www.rfc-editor.org/info/rfc5996>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<http://www.rfc-editor.org/info/rfc7401>>.

## Authors' Addresses

Robert Moskowitz  
Huawei  
Oak Park, MI 48237  
USA

Phone: +1-248-968-9809  
Email: [rgm@htt-consult.com](mailto:rgm@htt-consult.com)

Jinwei Xia  
Huawei  
101 Software Avenue  
Nanjing, Yuhua District 210012  
China

Phone: +86-025-84565890  
Email: [xiajinwei@huawei.com](mailto:xiajinwei@huawei.com)

DOTS  
Internet-Draft  
Intended status: Standards Track  
Expires: August 22, 2016

K. Nishizuka  
NTT Communications  
L. Xia  
J. Xia  
Huawei Technologies Co., Ltd.  
D. Zhang

L. Fang  
Microsoft  
February 19, 2016

Inter-domain cooperative DDoS protection problems and mechanism  
draft-nishizuka-dots-inter-domain-mechanism-00

## Abstract

As DDoS attack evolves rapidly in the aspect of volume and sophistication, cooperation among operators for sharing the capacity of the protection system to cope with it becomes very necessary. This document describes some possible solutions to the cooperative inter-domain DOTS problems.

## Status of This Memo

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

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

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

This Internet-Draft will expire on August 22, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Motivations . . . . .	2
2. Cooperative DDoS Protection Problems . . . . .	3
2.1. Bootstrapping Problem . . . . .	3
2.1.1. Automatic Provisioning vs Manual Provisioning . . . . .	3
2.2. Coordination Problem . . . . .	4
2.3. Near Source Protection Problem . . . . .	4
2.4. Returning Path Problem . . . . .	5
2.5. Billing Information Problem . . . . .	5
3. Inter-domain DOTS Architecture . . . . .	5
3.1. Distributed Architecture . . . . .	6
3.2. Centralized Architecture . . . . .	9
4. Inter-domain DOTS Protocol . . . . .	10
4.1. Provisioning Stage . . . . .	11
4.1.1. Messages . . . . .	12
4.1.2. Operations . . . . .	14
4.2. Signaling Stage . . . . .	14
4.2.1. Messages . . . . .	15
4.2.2. Operations . . . . .	20
5. Security Considerations . . . . .	21
6. IANA Considerations . . . . .	21
7. Normative References . . . . .	21
Authors' Addresses . . . . .	22

## 1. Motivations

These days, DDoS attacks are getting bigger and more sophisticated. Preliminary measures for minimizing damages caused by such attacks are indispensable to all organizations facing to the internet. Due to the variations of UDP reflection attack, there are still too big platforms of DDoS attack which consist of vulnerable servers, broadband routers and other network equipments distributed all over the world. Because of the amplification feature of the reflection attack, attackers can generate massive attacks with small resources. Moreover, there are many booters who are selling DDoS attacks as a service. DDoS attack is commoditized, so frequency of DDoS attack is also increasing.

These trends of the attack could exceed a capacity of a protection system of one organization in the aspect of volume and frequency.



Therefore, sharing the capacity of the protection system with each other to cope with such attacks becomes very necessary.

By utilizing other organization's resources, the burden of the protection is shared. The shared resources are not only CPU/memory resources of dedicated mitigation devices but also the capability of blackholing and filtering. We call the protection which utilize resources of each other "cooperative DDoS protection".

The cooperative DDoS protection have numerous merits. First, as described above, it can leverage the capacity of the protection by sharing the resources among organizations. Generally DDoS attack happens unexpectedly, thus the capacity utilization ratio of a protection system is not constant. So, while the utilization ratio is low, it can be used by other organization which is under attack. Second, organizations can get various countermeasures. If an attack is highly sophisticated and there is no countermeasure in the system, cooperative DDoS protection can offer optimal countermeasure of all partners. Third, it can block malicious traffic near to the origin of the attack. Near source defense is ideal for the health of the internet because it can reduce the total cost of forwarding packets which are mostly consist of useless massive attack traffic. Moreover, it is also very effective to solve the inter-domain uplink congestion problem. Finally, it can reduce time to respond. After getting attacked, prompt response is important because the outage of the service can make significant loss to the victim organization. Cooperating channel between partner organizations would be automated by dots protocol.

## 2. Cooperative DDoS Protection Problems

In this section, problems regarding to cooperative DDoS protection are described.

### 2.1. Bootstrapping Problem

DDoS attacks are unpredictable, so preliminary measures are important to maximize the utility of cooperative DDoS protection, which are accomplished by provisioning of DDoS protection system of each other in advance. However, it is difficult to set up DDoS protection of each other's service in secure manner.

#### 2.1.1. Automatic Provisioning vs Manual Provisioning

Manual provisioning is easier way to utilize DDoS protection service of other organizations. An organization can trust other organization who are going to use their DDoS protection service by any means like phone, e-mail, Web portal, etc,. However, it will take much time to

provision the DDoS protection system, then the attack will succeed to make significant impact on the protected service. To reduce the time to start the protection, automatic provisioning is desirable. If an organization could acquire relevant information of the DDoS protection service of other organization and utilize it by dots signaling in short time, the cooperative DDoS protection will succeed at a certain level. It is needed to find a way to provision other DDoS protection service in secure manner. In the later section, the total scenario is divided into two stages, those are provisioning stage and signaling stage. It is assumed that dots signaling is authorized by some credentials provided in provisioning stage in advance. Other important works carried out in the bootstrapping process are auto-discovery, automatic capability building between the member DDoS protection service providers as the basis for the following coordination process.

## 2.2. Coordination Problem

The number of the member DDoS protection service provider of cooperative DDoS protection is important factor. If only two providers are involved, there is bilateral relationship only. It is easy to negotiate about the capacity of their own DDoS protection system. In the state of emergency, they can decide to ask for help each other if the capacity of their own system is insufficient. When a lot of providers are joining cooperative DDoS protection, it is difficult to decide where to ask for help. They need to negotiate about their capacity with every participant. It is needed to take into account all combinations to do appropriate protection. The coordination between the member providers of cooperative DDoS protection is a complete process consisting of mitigation start/stop, status notification, mitigation policy updates and so on.

In addition, inter-domain uplink congestion problem can only be solved by coordinating the protection services provided by the upstream operators.

## 2.3. Near Source Protection Problem

Stopping malicious traffic at the nearest point in the internet will reduce exhaustion of resources in all the path of the attack. To find the entry point of the attack, traceback of the attack traffic to the origin is needed. If there is cooperative partner near the attack source, asking for help to the ISP is most effective. However, the problem is that it is difficult to decide which ISP is nearest to the attack source because in many cases source address of attack packets are spoofed to avoid to be visible from others. Moreover, some topology information of ISP's NW will be uncovered in order to make the decision correctly, however there could be privacy

protection issue between ISPs. Those problems will lead to the difficulties of locating the attack source. The problems can be divided into two problems. The first is how to find the attacker. The second is how to decide whom to ask for help.

#### 2.4. Returning Path Problem

As one of protection methods, some DDoS protection service provider announce BGP route to detour the attack traffic to their own network to deal with it. After scrubbing, cleaned traffic should be returned to the original destination. The returning path is often called "clean pipe". The DDoS service provider should be careful about routing loop because if the end point of a clean pipe is still included in a reach of the announced BGP route, the traffic will return to the mitigation path again and again. When thinking about cooperative DDoS protection, returning path information should be propagated to partners.

#### 2.5. Billing Information Problem

This is not technical nor a part of dots protocol but it has relation to deployment models. If other organization utilized resources of DDoS protection service, it is natural to charge it according to the amount of use. However, how to count the amount of use differs among DDoS protection service providers. For example, some DDoS protection service provider charges users by volume of the attack traffic or dropped packets. On the other hand, some of them use volume of normal traffic. Number of execution can be also used. We can not decide what information should be taken into account for billing purpose in advance, however those information is needed to be exchanged while coordinating DDoS protection. These information could be also used to determine which service would be used when asking for help. Though it is out of the scope of dots, coordinating and optimizing the cooperation in the aspect of business is difficult to solve.

### 3. Inter-domain DOTS Architecture

As described above, with the fast growth of DDoS attack volume and sophistication, a global cooperative DDoS protection service is desirable. This service can not only address the inter-domain uplink congestion problem, but also take full advantage of global DDoS mitigation resources from different ISPs efficiently and enable the near source mitigation. Moreover, with the way of providing DDoS mitigation as service, more customers will get it flexibly by their demands with maximized territory and resources. Together with on-premise DDoS protection appliance, the multiple layer DDoS system provides a comprehensive DDoS protection against all types of

attacks, such as application layer attacks, network layer large traffic attacks and others. The signaling mechanisms between on-premise DDoS protection appliance and cloud service are in scope of DOTS.

The inter-domain DDoS protection service is set up based on the member ISPs' own DDoS protection systems and the coordination protocol between them. The inter-domain protocol (or signaling mechanism) for the goal of DDoS protection coordination is the main focus of this document. Note that not only ISPs but also cloud based DDoS protection providers can participate in the inter-domain DDoS protection service. In general, the member ISP's own DDoS systems should at least consist of controller, mitigator or possibly flow analyser, which:

controller: be responsible for intra-domain DDoS mitigation  
controlling and communication for customers and inter-domain  
coordination

mitigator: be responsible for mitigation and results report

flow analyser: be responsible for attack detection and source  
traceback.

The inter-domain DDoS protection service has two different deployment models: distributed architecture or centralized architecture. The following parts give the respective discussion to them by aligning to DOTS terms.

### 3.1. Distributed Architecture

Several ISPs can set up the bilateral cooperative relation of DDoS protection between each other, thereby a distributed inter-domain DDoS protection service is provided with the support of peer to peer communication. The corresponding distributed architecture is illustrated in the following diagram:

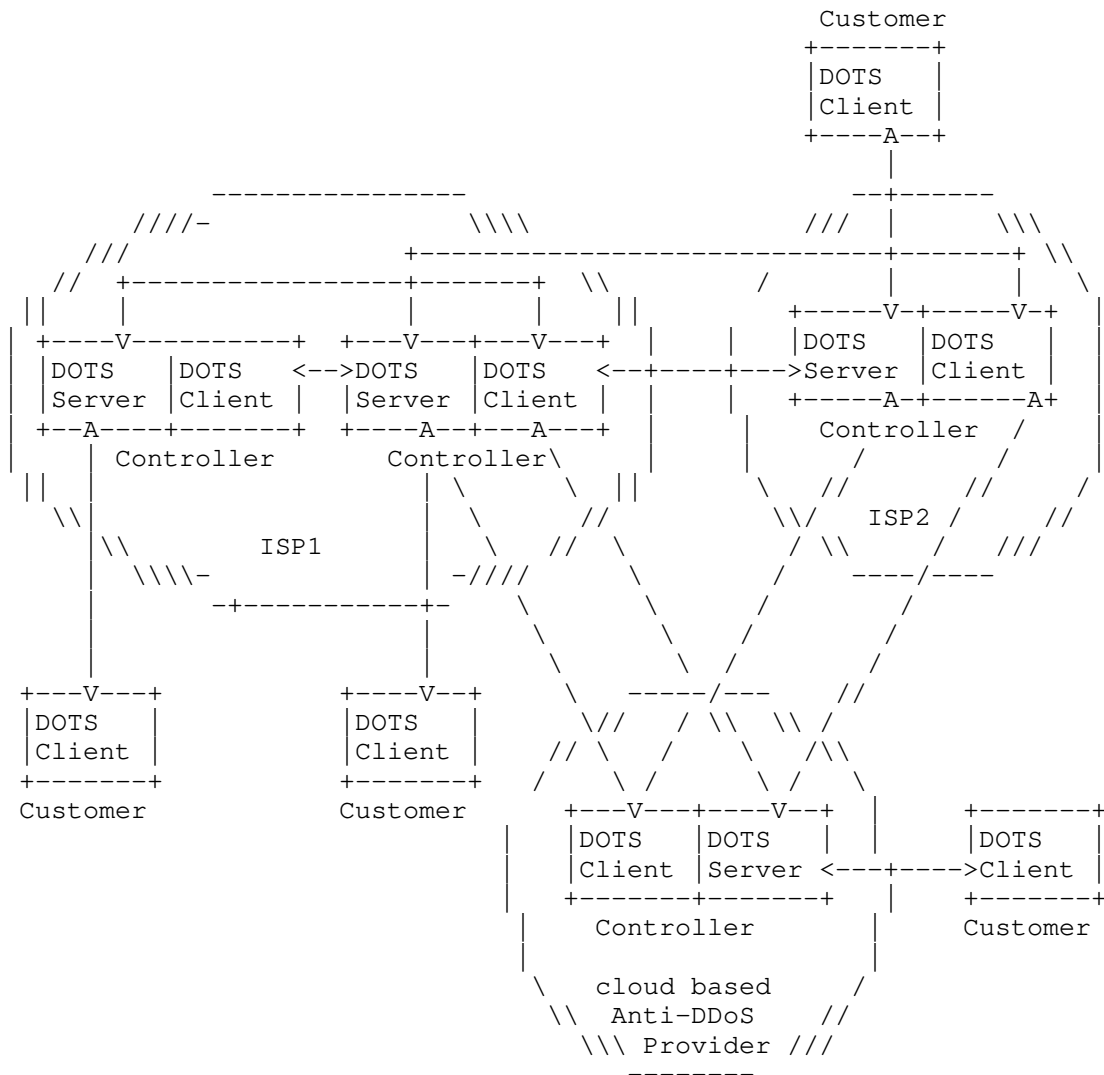


Figure 1: Distributed Architecture for Inter-domain DDoS Protection Service

As illustrated in the above diagram, when the customer is suffering a large traffic DDoS attack, it acts as the DOTS client to request DDoS protection service from its ISP. The ISP controller acts as the DOTS server to authenticate the customer's validity and then initiate the intra-domain DDoS mitigation service with its own resource for the customer. If the ISP controller finds the attack volume exceeds its capacity, or the attack type is unknown type, or its inter-domain

upstream link is congested, it should act as the DOTS client to request inter-domain coordination to all or its upstream ISP controllers which it has cooperative relation with. The ISP controller should support the functions of DOTS server and DOTS client at the same time in order to participate in the system of inter-domain DDoS protection service. In other words, as the representative for an ISP's DDoS protective service, the ISP controller manages and provides DDoS mitigation service to its customer in one hand, but may require helps from other ISPs under some situation especially when the attack volume exceeds its capacity or the attack is from other ISPs. The inter-domain coordination can be a repeated process until the attack source faced ISP receives the inter-domain coordination request and mitigates the attack traffic.

In particular, each ISP is able to decide its responding actions to its peering ISPs' request flexibly by following the internal policies, such as whether or not perform the mitigation function, or whether or not relay the request message to other ISPs. But these are out of the scope of this document.

The distributed architecture is straightforward and simple when the member ISPs are not too many. Regarding to deployment, all the work an ISP needs to do is to configure other cooperative member ISPs' information (i.e., IP, port, certificate, etc) and relevant cooperative policies for the following inter-domain communication. Regarding to operation, each ISP's controller just performs the mitigation service according to customer's request and possibly asks for inter-domain helps to other ISPs if necessary. In the meantime, the mitigation report and statistics information is required to exchange between the peering ISPs for the goal of monitoring and accounting.

But there are still some problems for the distributed architecture:

- o Every ISP controller only has the information of those ISPs which have cooperative relation with it, not the all ISPs participated in the inter-domain DDoS protection service. The incomplete information may not lead to the most optimized operation
- o When the member ISPs reach a certain number, a new joining ISP will be required to configure and maintain a lot of peering ISPs' information. It's complex and error-prone
- o Due to the exclusive repeated nature of the this architecture mentioned above, it's possible that the really effective mitigation service by one upstream ISP happens after several rounds of repeating the inter-domain coordination process. It may take a long time and is unacceptable.

### 3.2. Centralized Architecture

For the centralized architecture, the biggest difference from the distributed architecture is that a centralized orchestrator exists aimed at controlling the inter-domain DDoS coordination centrally. The centralized architecture for the inter-domain DDoS protection service is illustrated in the following diagram:

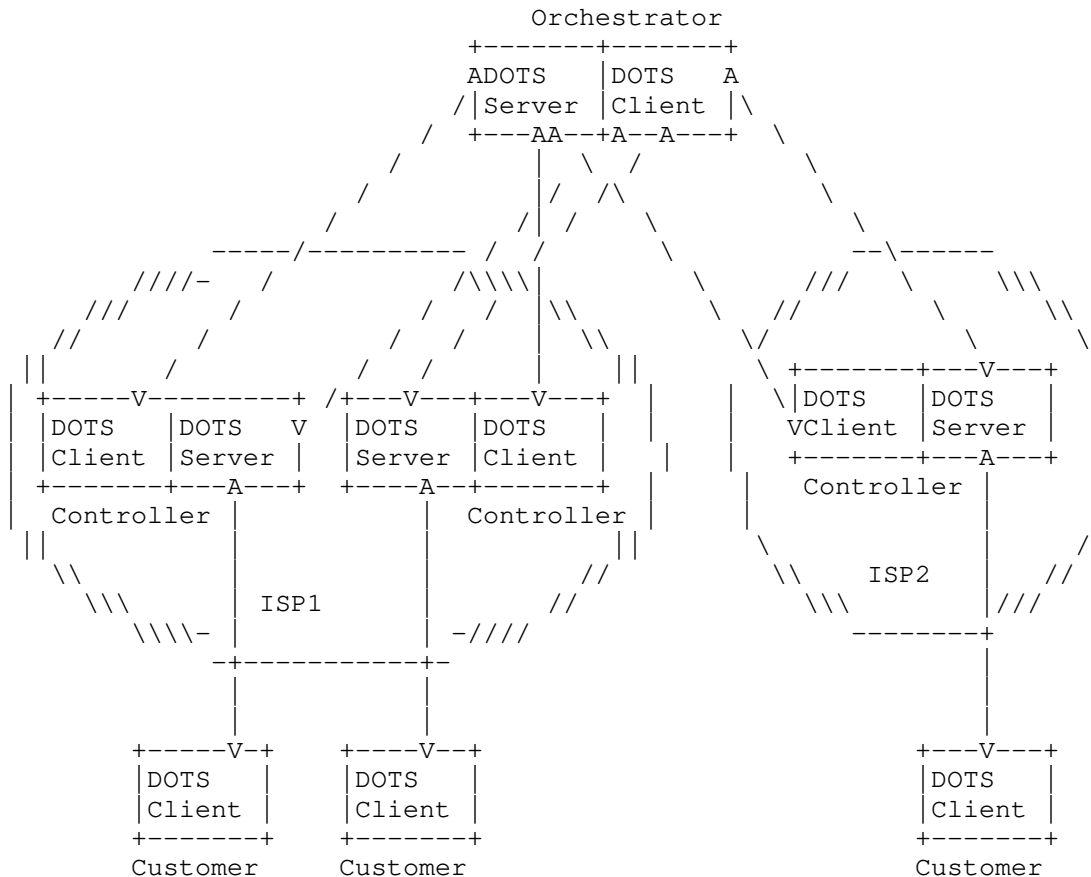


Figure 2: Centralized Architecture for Inter-domain DDoS Protection Service

As illustrated in the above diagram, the orchestrator is the core component to the inter-domain system. Each ISP controller only communicates with it for the goal of registering, coordination requesting and reporting. When it receives the inter-domain coordination request message from the ISP controller, a simple way is to notify all the other ISP controllers which have registered to the

orchestrator, to enable the possible mitigation services. Another way is to choose a number of ISPs to notify them enable the mitigation services according to the traceback result or other policies. The details is to be added in future. Based on the above analysis, the orchestrator is also a combination of DOTS server and DOTS client which support both functions at the same time.

In addition to the orchestrator and its related functions, the signaling and operations of centralized architecture are very similar to the implementation of distributed architecture.

The centralized architecture has its own characteristics as below:

- o Due to the centralized architecture, the orchestrator is easy to suffer the problems of congestion or performance downgrade to influence the availability of the whole system. This can be improved by the redundant orchestrator deployment
- o A centralized orchestrator facilitates the auto-discovery mechanism for the member ISPs. And for each ISP controller, its deployment and operation becomes very easy cause it is only required to communicate with the orchestrator during the whole time
- o With the help of direct communication between the orchestrator and all ISP controllers, an inter-domain DDoS coordination is finished in a short and fixed time period.

#### 4. Inter-domain DOTS Protocol

According to [I-D.draft-ietf-dots-requirements], DOTS protocols MUST take steps to protect the confidentiality, integrity and authenticity of messages sent between the DOTS client and server, and provide peer mutual authentication between the DOTS client and server before a DOTS session is considered active. The DOTS agents can use HTTPS (with TLS) for the goal of protocol security. The HTTP RESTful APIs are used in this section as the protocol channel, and the DOTS message content can be in JSON format.

With respect to the inter-domain DOTS protocol, all the DOTS messages are exchanged between DOTS client and server, no matter what the architecture (distributed or centralized) is. So, the message formats and operations of DOTS protocol is ought to be unified for all architecture options. The DOTS messages can be categorized by which stage they are mainly required in during DDoS protection, as below:



- o Provisioning stage: Before getting attacked by malicious traffic, a DOTS client needs register to the DOTS server, as well as enable capacity building in advance;
- o Signaling stage: This stage covers the time period when the DDoS attack is happening. At the beginning, the DOTS client should signal the DOTS server to provide DDoS mitigation service to the customer service under attack. At the end, once the attack is over, the DOTS client should notify the DOTS server to stop the mitigation service.

DOTS protocol can run on HTTPS (with TLS) and employ several different ways for authentication:

- o Employ bidirectional certificate authentication ([ITU-T X.509]) on the DOTS server and client: Both DOTS server and client need to verify the certificates of each other;
- o Employ unidirectional certificate authentication ([ITU-T X.509]) on the DOTS server: Only the DOTS server needs to install the certificate. The DOTS client needs to verify its certificate. In the opposite direction, DOTS server can authenticate DOTS client by the ways of user/role:password, IP address white-list or digital signature;
- o Employ bidirectional digital signature authentication on the DOTS server and client: In this condition, the DOTS server and client must keep the customer's private key safely, which is used for calculate the digital signature.

Besides authenticating the DOTS client, the DOTS server also verifies the timestamp of the packets from the DOTS client. If the time difference between the timestamp and the current time of the DOTS server exceeds the specified threshold (60 seconds as an example), the DOTS server will consider the packet invalid and will not process it. Therefore, NTP must be configured on both the DOTS server and client to ensure time synchronization. This method can protect DOTS server against the replay attack effectively.

The following sections present the detailed description of all the DOTS messages for each stage, and the relevant DOTS protocol operations.

#### 4.1. Provisioning Stage

In the provisioning stage, DOTS client can be located in the customer side, in the ISP controller or in the inter-domain orchestrator (for the centralized architecture). In any cases, the DOTS client is

required to register to its peering DOTS server which provides the intra/inter domain DDoS mitigation service to it, in order to set up the DOTS protocol channel. More importantly, the registration process also facilitates the auto-discovery and capacity building between the DOTS client and server.

#### 4.1.1. Messages

In the provisioning stage, the messages of registration (DOTS client to server), registration response (DOTS server to client), registration cancelling (DOTS client to server) and registration cancelling response (DOTS server to client) are required.

The HTTP POST method with the message body in JSON format is used for the registration and registration response messages as below:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/registration
registration body:
```

```
{
  "customer_name": string;
  "ip_version": string;
  "protected_zone": string;
  "protected_port": string;
  "protected_protocol": string;
  "countermeasures": string;
  "tunnel_information": string;
  "next_hop": string;
  "white_list": string;
  "black_list": string;
}
```

```
registration response body:
```

```
{
  "customer_name": string;
  "customer_id": string;
  "access_token": string;
  "thresholds_bps": number;
  "thresholds_pps": number;
  "duration": number;
  "capable_attack_type": string;
  "registration_time": string;
  "mitigation_status": string;
}
```

Registration body:

customer\_name: The name of the customer (DOTS client);  
ip\_version: Current IP version. It can be "v4" or "v6";  
protected\_zone: Limit the address range of protection.

Especially it will be limited to the prefixes possessed by the customer;  
protected\_port: Limit the port range of protection;  
protected\_protocol: Valid protected protocol values include tcp and udp;  
countermeasures: Some of the protection need mitigation and others need Blackholing;  
tunnel\_information: The tunnel between the mitigation provider's network and the customer's network. Tunnel technologies such as GRE[RFC2784] can be used to return normal traffic;  
next\_hop: The returning path to the customer's network;  
white\_list: The white-list information provided to the DOTS server;  
black\_list: The black-list information provided to the DOTS server.

registration response body:  
customer\_name: The name of the customer (DOTS client);  
customer\_id: The unique id of the customer (DOTS client);  
access\_token: Authentication token (e.g. pre-shared nonce);  
thresholds\_bps: If an attack volume is over this threshold, the controller will reject the protection in order to compliance with the negotiated contract;  
thresholds\_pps: If an attack volume is over this threshold, the controller will reject the protection in order to compliance with the negotiated contract;  
duration: If an attack longed over this threshold, the controller will reject the protection in order to compliance with the negotiated contract;  
capable\_attack\_type: Limit the protectable attack type;  
registration\_time: The time of registration;  
mitigation\_status: The status of current mitigation service of the ISP.

Similarly, another HTTP POST method with the message body in JSON format is used for the registration cancelling and registration cancelling response messages as below:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                    registration_cancelling
registration cancelling body:
{
  "customer_id": string;
  "reasons": string;
}
registration cancelling response body:
{
  "customer_id": string;
  "result": string;
}
```

Registration cancelling body:  
customer\_id: The unique id of the customer (DOTS client);  
reasons: The reasons why the DOTS client cancel the registration;

registration cancelling response body:  
customer\_id: The unique id of the customer (DOTS client);  
result: The final result if the DOTS controller accepts  
the registration cancelling request.

#### 4.1.2. Operations

The main operations in the provisioning stage include:

- o The customers (DOTS client) registers to ISP controller with capability building including protection methods, process capacity, ip address scope, deployment position, etc;
- o The DOTS client in ISP controller registers to the DOTS server in inter-domain orchestrator (centralized architecture) or other ISP controllers (distributed architecture) according to inter-domain DDoS protection requirements;
- o The DOTS client can send the registration cancelling message to the DOTS server for cancelling its DDoS protection service.

#### 4.2. Signaling Stage

Once the DOTS client detects the attack to the customer service, a mitigation request message is created and sent to the provisioned DOTS server to call for the ISP DDoS protection service. The DOTS server decides to protect the customer service based on the provisioned information, and sends the mitigation response message to the DOTS client. One ISP's DOTS server may resume sending the mitigation request message to other ISPs' DOTS server to request the inter-domain coordinated mitigation service while it notices it isn't

able to handle the attack by itself. Meanwhile, some other messages are required for status exchange and statistics report. When the DOTS server is informed from the mitigator that the attack is over, it should notify the DOTS client to terminate the mitigation service.

#### 4.2.1. Messages

In the signaling stage, the messages of mitigation request (DOTS client to server), mitigation response (DOTS server to client), mitigation scope update (DOTS client to server), mitigation efficacy notification (DOTS client to server), mitigation status request (DOTS client to server), mitigation termination notification (DOTS client to server), mitigation termination response (DOTS server to client) and heartbeat (bidirectional message) are required.

##### Mitigation Request:

A HTTP POST method with the message body in JSON is used for the mitigation request and response messages:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                mitigation_request
mitigation request body:
{
  "access_token": string;
  "traffic_protocol": string;
  "source_port": string;
  "destination_port": string;
  "source_ip": string;
  "destination_ip": string;
  "time": string;
  "dstip_current_bps": string;
  "dstip_current_pps": string;
  "dstip_peak_bps": string;
  "dstip_peak_pps": string;
  "dstip_average_bps": string;
  "dstip_average_pps": string;
  "bandwidth_threshold": string;
  "type": string;
  "severity": string;
  "mitigation_action": string;
}
mitigation response body:
{
  "access_token": string;
  "mitigation_id": number;
  "policy_id": number;
```

```
"description": string;  
"start_time": string;  
"current_bps": string;  
"current_pps": string;  
}
```

mitigation request body:

```
access_token: Authentication token (e.g. pre-shared nonce);  
traffic_protocol: Valid protocol values include tcp and udp;  
source_port: For TCP or UDP or SCTP or DCCP:  
the source range of ports (e.g., 1024-65535);  
destination_port: For TCP or UDP or SCTP or DCCP:  
the destination range of ports (e.g., 1-443);  
source_ip: The source IP addresses or prefixes;  
destination_ip: The destination IP addresses or prefixes;  
time: the time the event was triggered. The timestamp of  
the record may be used to determine the resulting duration;  
dstip_current_bps: The current volume of the attack in bps;  
dstip_current_pps: The current volume of the attack in pps;  
dstip_peak_bps: The peak volume of the attack in bps;  
dstip_peak_pps: The peak volume of the attack in pps;  
dstip_average_bps: The average volume of the attack in bps;  
dstip_average_pps: The average volume of the attack in pps;  
bandwidth_threshold: Event bandwidth as a % of overall  
link capacity of DOTS client;  
type: The attack type determined from the attack definitions;  
severity: The severity of the attack;  
mitigation_action: The mitigation actions customer anticipated,  
such as: block, mitigation, etc.
```

mitigation response body:

```
access_token: Authentication token (e.g. pre-shared nonce);  
mitigation_id: The unique mitigation event identifier;  
policy_id: Protection policy identifier allocated in  
the DOTS server;  
description: Textual notes;  
start_time: The time the mitigation was started.  
current_bps: The current level of offramped traffic in bps;  
current_pps: The current level of offramped traffic in pps.
```

Mitigation Status Exchange:

A HTTP POST method with the message body in JSON is used for the mitigation scope update and response message:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                    mitigation_scope_update
mitigation scope update body:
{
    TBD
}
mitigation scope update response body:
{
    TBD
}

mitigation scope update body:
TBD

mitigation scope update response body:
TBD
```

A HTTP POST method with the message body in JSON is used for the mitigation efficacy notification message:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                    mitigation_efficacy_notification
mitigation efficacy notification body:
{
    TBD
}
mitigation efficacy notification response body:
{
    TBD
}

mitigation efficacy notification body:
TBD

mitigation efficacy notification response body:
TBD
```

A HTTP POST method with the message body in JSON is used for the mitigation status request message:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                    mitigation_status_request
mitigation status request body:
{
    "mitigation_id": number;
    "start_time": string;
    "end_time": string;
}
```

mitigation status request response body:

```
{
  "mitigation_id": number;
  "mitigation_status": number;
  "source_port": string;
  "destination_port": string;
  "source_ip": string;
  "destination_ip": string;
  "TCP_flag": string;
  "start_time": string;
  "end_time": string;
  "error_num": number;
  "routing_state": string;
  "forwarded_total_packets": number;
  "forwarded_total_bits": number;
  "forwarded_peak_pps": number;
  "forwarded_peak_bps": number;
  "forwarded_average_pps": number;
  "forwarded_average_bps": number;
  "malicious_total_packets": number;
  "malicious_total_bits": number;
  "malicious_peak_pps": number;
  "malicious_peak_bps": number;
  "malicious_average_pps": number;
  "malicious_average_bps": number;
  "record_time": string;
}
```

mitigation status request body:

mitigation\_id: The unique mitigation event identifier;  
start\_time: The requested start time for the duration  
of the mitigation status message;  
end\_time: The requested end time for the duration  
of the mitigation status message;

mitigation status request response body:

mitigation\_id: The unique mitigation event identifier;  
mitigation\_status: Current mitigation status,  
such as: pending, ongoing, done;  
source\_port: For TCP or UDP or SCTP or DCCP: the source  
range of ports (e.g., 1024-65535) of the discarded traffic;  
destination\_port: For TCP or UDP or SCTP or DCCP: the  
destination range of ports (e.g., 1-443) of the discarded traffic;  
source\_ip: The source IP addresses or prefixes of  
the discarded traffic;  
destination\_ip: The destination IP addresses or prefixes  
of the discarded traffic;  
TCP\_flag: TCP flag of the discarded traffic;



start\_time: The start time for the duration of this mitigation status message;  
end\_time: The end time for the duration of this mitigation status message;  
error\_num: error message id;  
routing\_state: Current routing state;  
forwarded\_total\_packets: The total number of packets forwarded;  
forwarded\_total\_bits: The total bits for all the packets forwarded;  
forwarded\_peak\_pps: The peak pps of the traffic forwarded;  
forwarded\_peak\_bps: The peak bps of the traffic forwarded;  
forwarded\_average\_pps: The average pps of the traffic forwarded;  
forwarded\_average\_bps: The average bps of the traffic forwarded;  
malicious\_total\_packets: The total number of malicious packets;  
malicious\_total\_bits: The total bits of malicious packets;  
malicious\_peak\_pps: The peak pps of the malicious traffic;  
malicious\_peak\_bps: The peak bps of the malicious traffic;  
malicious\_average\_pps: The average pps of the malicious traffic;  
malicious\_average\_bps: The average bps of the malicious traffic;  
record\_time: The time the mitigation status message is created;

#### Mitigation Termination:

A HTTP POST method with the message body in JSON is used for the mitigation termination notification message:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/  
mitigation_termination_notification  
mitigation termination notification body:  
{  
  "mitigation_id": number;  
  "reason": string;  
}
```

mitigation termination notification body:  
mitigation\_id: The unique mitigation event identifier;  
reason: The reason of notifying the DOTS client to terminate the mitigation service;

A HTTP POST method with the message body in JSON is used for the mitigation termination and response messages:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/
                    mitigation_termination
mitigation termination body:
{
  "mitigation_id": number;
}
mitigation termination response body:
{
  "mitigation_id": number;
  "start_time": string;
  "end_time": string;
}

mitigation termination body:
mitigation_id: The unique mitigation event identifier;

mitigation termination response body:
mitigation_id: The unique mitigation event identifier;
start_time: The start time of the mitigation service;
end_time: The end time of the mitigation service.
```

Heartbeat:

A HTTP POST method with the message body in JSON is used for the heartbeat message:

```
METHOD:POST - URL:{scheme}://{host}:{port}/dots/api/heartbeat
heartbeat body
{
}
```

#### 4.2.2. Operations

The main operations in the signaling stage include:

- o The customer (DOTS client) detects malicious attack, requests mitigation service to its ISP controller (DOTS server);
- o ISP controller authenticates the customer and provides its intra-domain mitigation service to customer;
- o When the ISP controller are mitigating the attack and finding the attack volume exceeds its capacity, or the attack type is unknown type, or its upstream link is congested, it should request to the inter-domain orchestrator for inter-domain cooperation;
- o The inter-domain orchestrator straightforwardly forward the mitigation request to all other registered ISP controllers to

enable possible mitigation services. It is simple and for avoiding privacy exposure of ISPs;

- o Working ISP controllers reports its statistics result by mitigation status request message to the orchestrator for counting purpose;
- o The customer can update its mitigation scope to the ISP controller. It also can notify its mitigation efficacy result to the ISP controller;
- o When the ISP controller is informed from the mitigator that the attack is over, it should notify the customer to terminate the mitigation service;
- o The heartbeat message is exchanged between the DOTS client and DOTS server to check their respective status. If any side of the channel fails to receive the heartbeat message, then it will trigger an alert or further investigation into why they never reached their destination.

## 5. Security Considerations

TBD

## 6. IANA Considerations

No need to describe any request regarding number assignment.

## 7. 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>>.
- [RFC2784] D. Farinacci., T. Li., S. Hanks., D. Meyer., and P. Traina., "Generic Routing Encapsulation (GRE), March 2000".
- [I-D.draft-ietf-dots-requirements] A. Mortensen., R. Moskowitz., and T. Reddy., "DDoS Open Threat Signaling Requirements, draft-ietf-dots-requirements-00, October 2015".

[I-D.draft-reddy-dots-transport]

T. Reddy., D. Wing., P. Patil., M. Geller., M.  
Boucadair., and R. Moskowitz., "Co-operative DDoS  
Mitigation, October 2015".

Authors' Addresses

Kaname Nishizuka  
NTT Communications  
GranPark 16F  
3-4-1 Shibaura, Minato-ku, Tokyo  
108-8118, Japan

EMail: kaname@nttv6.jp

Liang Xia  
Huawei Technologies Co., Ltd.  
101 Software Avenue, Yuhuatai District  
Nanjing, Jiangsu  
210012, China

EMail: frank.xialiang@huawei.com

Jinwei Xia  
Huawei Technologies Co., Ltd.  
101 Software Avenue, Yuhuatai District  
Nanjing, Jiangsu  
210012, China

EMail: xiajinwei@huawei.com

Dacheng Zhang  
Beijing  
China

EMail: dacheng.zdc@aliabab-inc.com

Luyuan Fang  
Microsoft  
15590 NE 31st St  
Redmond, WA 98052

EMail: lufang@microsoft.com

DOTS  
Internet-Draft  
Intended status: Informational  
Expires: September 21, 2016

K. Nishizuka  
NTT Communications  
March 20, 2016

Inter-Domain DOTS Use Cases  
draft-nishizuka-dots-inter-domain-usecases-01

Abstract

This document describes inter-domain use cases of the DDoS Open Threat Signaling(DOTS).

Status of This Memo

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

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

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

This Internet-Draft will expire on September 21, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Inter-Domain DDoS Protection Scenario . . . . .	3
3.1. Protection Methods . . . . .	4
3.1.1. Blackholing . . . . .	4
3.1.2. Selective Blackholing . . . . .	5
3.1.3. RTBH with uRPF . . . . .	5
3.1.4. BGP flowspec . . . . .	6
3.1.5. Filtering(ACL) . . . . .	6
3.1.6. DDoS mitigation Appliances . . . . .	7
3.1.7. Detouring Technologies . . . . .	8
3.2. Restriction on the Range of IP Addresses . . . . .	8
3.3. Attack Telemetry . . . . .	8
3.4. DDoS Protection Status . . . . .	11
3.5. DDoS Protection Registration . . . . .	11
4. Inter-Domain Dots Use Cases . . . . .	12
4.1. Customer-to-Provider Cases . . . . .	13
4.1.1. Usecase 1: Single-home Model . . . . .	13
4.1.2. Usecase 2: Multi-home Model . . . . .	13
4.2. Provider-to-Provider Cases . . . . .	15
4.2.1. Usecase 3: Delegation Model . . . . .	15
4.2.2. Usecase 4: Distributed Architecture Model . . . . .	16
4.2.3. Usecase 5: Centralized Architecture Model . . . . .	18
5. Security Considerations . . . . .	19
6. IANA Considerations . . . . .	19
7. References . . . . .	19
7.1. Normative References . . . . .	19
7.2. URL References . . . . .	20
Author's Address . . . . .	20

## 1. Introduction

Maximum size of DDoS attack is increasing. According to a report from Cloudflare[Cloudflare], in 2013, over 300 Gbps DDoS attack against Spamhaus was observed which exploited DNS reflection mechanism to create massive attack with intention to overwhelm the capacity of the targeted system.

If this trend continued, the volume of DDoS attack will exceed preparable DDoS protection capability by one organization mostly in the aspect of cost. Moreover, possibility of DDoS attack is unpredictable, so it is not realistic that every organization prepare sufficient DDoS protection system.

This problem could be solved by sharing DDoS protection system over multi-organizations. We can share the burden of protection against

DDoS attack by inter-domain cooperation. To accomplish this goal, we need a framework which use common interface to call for protection.

In order to describe the mechanism of such a framework, use cases are classified into intra-domain use cases and inter-domain use cases. The focus of this draft is inter-domain use cases, which can be categorized to customer-to-provider cases and provider-to-provider cases.

1. intra-domain use cases(a DOTS client, a DOTS server and mitigators are in the same organization)
2. inter-domain use cases(a DOTS server and mitigators are in a different organization from a DOTS client)

By blocking DDoS attack with inter-domain cooperation, average usage of DDoS mitigation equipment will increase. This will leverage total capacity of DDoS protection system in all over the internet. With this mechanism, we can manage DDoS attacks which exceed the capacity of its own platform.

## 2. Terminology

Terminology and acronyms are inherited from [I-D.draft-ietf-dots-requirements]

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].

## 3. Inter-Domain DDoS Protection Scenario

In this inter-domain DDoS protection scenario, it is assumed that a service of an organization is being attacked over the internet and a DOTS client in the organization ask for help to one or more DOTS server in other organizations through signal channel between DOTS elements. Then DOTS server enables mitigation by communicating with mitigators in its domain by conveying information provided by the DOTS client. As noted in [I-D.draft-ietf-dots-requirements], A DOTS server may also be a mitigator. The request for help would be made in the case that a capacity of a protection system in the attacked organization is insufficient to protect the service. If an up-link connected to transit networks get congested due to the massive DDoS attack, there is no way to protect the service other than asking for help to upper transit providers or cloud-type of DDoS mitigation providers.

Especially in the case of inter-domain DDoS protection, it would be needed to care about protection capability of mitigators. The capability would be defined by possible protection methods taken by the mitigator and restriction of the usage.

### 3.1. Protection Methods

There are many available protection methods of mitigators, which include blackholing, ACLs, flowspec, dedicated DDoS appliances, etc. Required information for protection vary according to the protection methods. Some of information are mandatory, others are optional. Though the minimum information for protection is IP address of the system under attack, optional information would increase efficiency of the protection. Also, these methods have their own max capacity. This section enumerates possible protection methods. For future extensibility, DOTS protocol should be independent of these method. However, these protection methods would depict the protection scenario by describing mandatory information and optional information.

#### 3.1.1. Blackholing

Black-holing technique blocks DDoS attacks destined to a particular network by driving all traffic to a null interface on routers. In RTBH, Remotely Triggered Black-Holing[RFC3882], BGP announcement triggers black-holing in their network or neighbor networks by advertising routes with unreachable next-hop address or dedicated black-holing community. This technique results in that all traffic destined to the attacked network will be dropped on all ingress routers of the announced AS. This technique is widely used in ISPs and IXPs[draft-ietf-grow-blackholing-00].

RTBH can be used over eBGP peering, thus it inherently works in inter-domain manner by signaling over BGP. However, a victim organization doesn't always have eBGP peer to RTBH-enabled neighbor AS from which DDoS attack is coming. DOTS-enabled RTBH can help such scenario. In this scenario, a DOTS client ask for help to a DOTS server in a transit network. Then the DOTS server triggers RTBH in its network by announcing blackholing BGP routes.

mandatory information: Destination Address

RTBH works with destination IP address only, thus a mandatory information conveyed by the DOTS client to the DOTS server is IP address or prefix of the victim system.

As noted in the security consideration section of [RFC3882], eBGP customers might be able to blackhole a particular subnet using the



blackhole communities. Like that, a DOTS client can blackhole a particular subnet by sending DOTS message with arbitrary destination address, which can be another attack vector. To eliminate the risk, the range of valid IP address should be limited to the prefixes of the victim organization.

### 3.1.2. Selective Blackholing

In the case of blackholing, it stops the traffic destined to the service totally. In a way, the "denial" of service is successful. Selective blackhole provides the ability to limit the scope of the blackholing. It allows more flexible blackholing because some of traffic are blocked at the same time others are not affected according to geographic locations.

mandatory information: Destination Address

optional information: BGP Community, Next-hop Address

Selective Blackholing also works with destination IP address only. Moreover, by sending intended BGP community of selective blackholing, it gives more effective control on DDoS attacks. When some network element in the transit network announced the selective blackholing route, the next-hop address of the announced route should be unchanged from the original announcement because the traffic not blocked by selective blackholing still should be destined to the original network. Therefore, the DOTS server might need to know a desired next-hop of the prefix of the victim network.

### 3.1.3. RTBH with uRPF

Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF) [RFC5635] is expansion of destination-based RTBH filtering which enable filtering by source address.

By coupling unicast Reverse Path Forwarding (uRPF) [RFC3704] techniques with RTBH filtering, packets will be discarded not based on destination address, but on source address of DDoS traffic.

mandatory information: Source Address

By sending source address of the attack traffic from a DOTS client to a DOTS server, the DOTS server can utilize RTBH with uRPF via BGP. However, this technique also drops packets destined to other networks, which can be another denial-of-service of the source address depending on the topology.

#### 3.1.4. BGP flowspec

BGP flowspec[RFC5575] defines a new BGP NLRI encoding format by which routing system can propagate information regarding more specific components of the traffic. By pre-defined action rules, this technique can be used to automate inter-domain coordination of traffic filtering.

mandatory information:

- Flow Type:
  - Destination Prefix
  - Source Prefix
  - IP Protocol
  - Port
  - Destination Port
  - Source Port
  - ICMP type
  - ICMP Code
  - TCP flags
  - Packet Length
  - DSCP
  - Fragment
- Action Rule:
  - traffic-rate
  - traffic-action
  - redirect
  - traffic-remarking

Like blackholing, if a victim organization doesn't have capability of BGP flowspec, DOTS protocol could help it to work in inter-domain manner. If a DOTS client got a statistics of an ongoing attack to its site, it can send a combination of flow type and action rule information to a DOTS server in other network. Then the DOTS server can generate BGP flowspec route based on the information provided by the DOTS signaling, then it will be applied to its network to make it work.

#### 3.1.5. Filtering (ACL)

Access list (ACL) based filtering is widely used in ISPs to protect customers. They configure the routers connected to the customer manually or automatically to discard or rate-limit traffic base on the request from the customer. This kind of effort can be covered by DOTS protocol. Here is an example of the mandatory information of ACL.

mandatory information:

Match Rule:  
  IP Protocol  
  Destination Prefix  
  Source Prefix  
  Destination Port  
  Source Port  
Action Rule:  
  permit  
  deny

### 3.1.6. DDoS mitigation Appliances

There are many DDoS mitigation appliances, however, they have their own implementation and information model which result in that there is no compatibility each other. As noted in [draft-ietf-dots-use-cases], providing a standard-based mechanism is one of the goal of the DOTS. The merit of DDoS mitigation appliances is that only the malicious traffic will be discarded on the box and the scrubbed normal traffic will be returned to the original service, thus service continuity will be kept. Various countermeasures are implemented on those appliances to eliminate the possibility of false positives and false negatives. DDoS mitigation appliances can be used in intra-domain manner and inter-domain manner.

Some of ISPs are using DDoS mitigation appliances to protect their customer. If a mitigation box is placed inline to a customer and dedicated only to them, the customer would be always benefited from it. In this case, a DOTS client would send message to a DOTS server about only turning on/off of protection. A mitigation box can be placed on offramp position and shared with many customers because it is more cost effective. In this case, the mitigation can be accomplished by combined with detouring technologies. The DDoS mitigation appliances apply pre-defined countermeasures with the destination IP address of the targeted customer.

The total volume of processable traffic is limited to the capacity of the hardware. Therefore, if the DDoS mitigation appliances are shared among customers, capability should be negotiated carefully because insufficient capacity compared to total volume[bps/pps] of DDoS traffic could affect the service. Traffic volume and other attack telemetry can help the mitigation appliances to determine the mitigation behavior. Attack telemetry is noted in more detail in a later section.

mandatory information: Destination Address

optional information: (Desired)Countermeasures, Attack Telemetry

### 3.1.7. Detouring Technologies

Detouring technologies are used with other protection methods to deal with DDoS attack traffic in its domain. It eases topological constraints of protection methods and leverages limited capacity of them. By injecting more specific route in routing system, the attack traffic would be diverted to protection instance of mitigators. After the classification of malicious traffic and normal traffic, normal traffic should be returned to the original path, however simply returning traffic to the internet can cause routing loop because the returning traffic could re-enter the diversion path again. To avoid this routing loop, the safe returning path should be designated. If there is no dedicated line between the mitigator and the service, tunnel technology such as GRE[RFC2784] can be used. In that case, tunnel information should be provided. In general, next-hop and prefix information should be provided to the DOTS server to determine the returning path of the mitigated traffic.

mandatory information: Destination Address, Next-Hop

optional information: Tunnel Information

### 3.2. Restriction on the Range of IP Addresses

As reviewed in the previous section, some of protection methods can be another denial-of-service vector to other organization if there is no restriction on the range of destination IP addresses. Especially, in case of blackholing, they can abuse other systems by blocking all of the traffic. A DOTS server SHOULD refuse request from a DOTS client if it could result in packet loss of communication of third party.

### 3.3. Attack Telemetry

Attack telemetry is a set of summarized traffic information which characterizes the feature of the DDoS attack. Attack telemetry implicitly indicates the reason why the DOTS client assumed the observed traffic contains an attack. A DOTS client can call for help to a DOTS server by sending attack telemetry with authorization information via DOTS signal.

The DOTS server which received the DOTS signal reacts to start mitigation as follows:

1. The DOTS server checks the authorization information to decide the signaling is legitimate or not. If failed, it may return an error status.

2. The DOTS server checks the destination IP address in the request with according DDoS protection entity. If the IP address doesn't match for the prefixes of the customer who made the DOTS request, there might be a risk of packet loss of communication of third party, then it may return an error status.
3. The DOTS server selects an appropriate protection method while checking a protection capability of mitigators.
4. The DOTS server enables mitigation by communicating with the mitigator in its domain by conveying attack telemetry provided by the DOTS client.

The following list is an attack telemetry which characterizes the feature of the DDoS attack. As reviewed in the previous section, a destination IP address is key value which identifies a series of DDoS attack.

Attack Telemetry:

Mandatory:

Dst IP

Optional:

Attack ID

Dst Port

Src IP/Port

TCP Flag

Type of Attack

(Average/Maximum/Current)Traffic Volume[bps/pps]

Severity

Attack Start Time

Duration

o Attack ID

Attack ID could be assigned by a DOTS client. By receiving the attack ID, a DOTS server can tell the attack vector is the same or not from the observation of the DOTS client.

o Dst Port

Destination port of the DDoS attack characterizes the attack because it indicates what kind of service is targeted. This information can be used especially by filter type of protection methods. However, it should be noted that the targeted port can be changed by the attacker if they noticed the attack on the port is not effective.

o Src IP/Port

Source IP/Port of the DDoS attack characterizes the attack. Source port indicates the attack platforms in the case of amplification attack. Blocking or rate-limiting based on the source IP/Port can mitigate the attack effectively. However, in some cases, source IP/Port of the DDoS attack are spoofed. They could be widely spread in address space and continuously changing. Thus, the mitigation based on source IP/Port information is not always applicable.

- o TCP Flag

TCP flag of the DDoS attack characterizes the attack because it indicates the attack vector itself. TCP flag information can be used to distinguish malicious traffic and legitimate traffic.

- o Type of attack

Similar to TCP flag information, type of attack declares that what kind of attack vector is used by the attacker, ex) fragment attack, land attack etc,.. Decision of the type of attack might be overwritten by the mitigator if it can inspect the traffic more deeply.

- o (Average/Maximum/Current)Traffic Volume[bps/pps]

Traffic volume information can be used to determine protection method. However, In the case of massive DDoS attack, the circuit connected to the internet could be saturated by the traffic, so there is no way to know how much traffic is incoming on the saturated link from the victim network. Thus, traffic volume information provided by the DOTS client is optional information.

- o Severity

Severity information can be used to determine protection method. However, in many cases, DDoS attack vectors change time to time, so there is no constant index of severity. Moreover, the monitoring system on the service side could look through the important attack vector which is very severe to the service, so the severity must be overwritten by the mitigator if it can inspect the traffic more deeply. Therefore this is optional information.

- o Attack start time and Duration

Attack start time and Duration information indicates the status and the severity of the attack. The DOTS server and the mitigator can find the attack effectively by this information if it has a monitoring system in its domain.

### 3.4. DDoS Protection Status

The DOTS client may stop the mitigation by sending protection-stop-instruction message via DOTS protocol. However, sometimes, it is difficult to know whether the DDoS attack has ended or not from the monitoring point of the DOTS client especially in the inter-domain usecases. The information listed below should be provided from the DOTS server to the DOTS client.

- o Attack telemetry

Attack telemetry observed at the monitoring point of the DOTS server or the mitigator should be reported to the DOTS client periodically. In addition, the operator of the service will eager to know what kind of attack was attempted. Then, they can study how to try to find the best plan to cope with attacks in future.

- o Status of ongoing protection

Status of the protection(The attack is ongoing or not) will be used to determine that the system is already safe without the protection. The DOTS server should have interface from which the DOTS client can get the status of the protection.

- o Data for billing

In the inter-domain usecases, there might be a contract between two organizations. Some kind of data which indicates the usage of the protection resources may be used for billing. The typical examples are the number of the dropped packets, the number of the legitimate packets, duration of the protection, etc,.. Defining the billing data is out of scope of DOTS.

### 3.5. DDoS Protection Registration

If there is a contract between two organizations, a DOTS client might need to be registered to a DOTS server in advance. Authentication information might be provided in this registration. As reviewed in the previous section, some of protection methods needs more information in addition to attack telemetry in order to work properly. The information listed below might be registered in advance to the DOTS server, though these information could be registered and updated automatically during an attack.

- o Authentication information

Authentication information might be provided in customer registration. This information will be used in any phase after the registration to avoid abuse of the protection system.

- o Proper IP address

Proper IP address of the customer might be provided in customer registration. This information will be used by the DOTS server for checking a protection request in order to avoid abuse of the protection system. Also, consistency of the IP address might be checked with the routing system.

- o Desired Protection Method

A DOTS server will select a protection methods based on the attack telemetry provided by a DOTS client, however, the DOTS client could have preferred protection methods. If there is a possibility of misclassification on some protection method, the client might not choose it. The selectable protection methods might be registered to the DOTS server in advance.

- o Thresholds of Protection Methods

If a threshold of a protection, rate-limit for example, is stricter than a normal trend of the protected system, it may cause significant packet loss of the legitimate traffic. The appropriate thresholds of protection methods varies according to the customer's service. Thus, the customer might want to decide the thresholds of each protection method in advance.

- o Returning Path Information

If a protection method was coupled with detouring technologies, the legitimate traffic will be returned to the normal path to the customer. In order to make it work properly, the returning path information should be provided to the DOTS server in advance. Some protection method needs next-hop information and tunnel information.

#### 4. Inter-Domain Dots Use Cases

In inter-domain use cases, a DOTS server and mitigators are in a different organization from a DOTS client. Those can be categorized to customer-to-provider cases and provider-to-provider cases.



## 4.1. Customer-to-Provider Cases

## 4.1.1. Usecase 1: Single-home Model

The single-home model is the most basic model of the inter-domain usecase. There are one DOTS client in customer side and one DOTS server in provider side. The DOTS server communicate with the mitigator(s) in its domain to protect the service of the customer. If the service got attacked and the customer found suspicious traffic statistics, the DOTS client send attack telemetry, in which the IP address of the service under attack must be included, to the DOTS server via DOTS signaling. The DOTS server checks the message, then communicate with the mitigator in its domain to protect the service from attack traffic. The legitimate traffic will be kept going to the service. In the case of blackholing, all of the traffic destined to the service will be dropped in the provider's domain.

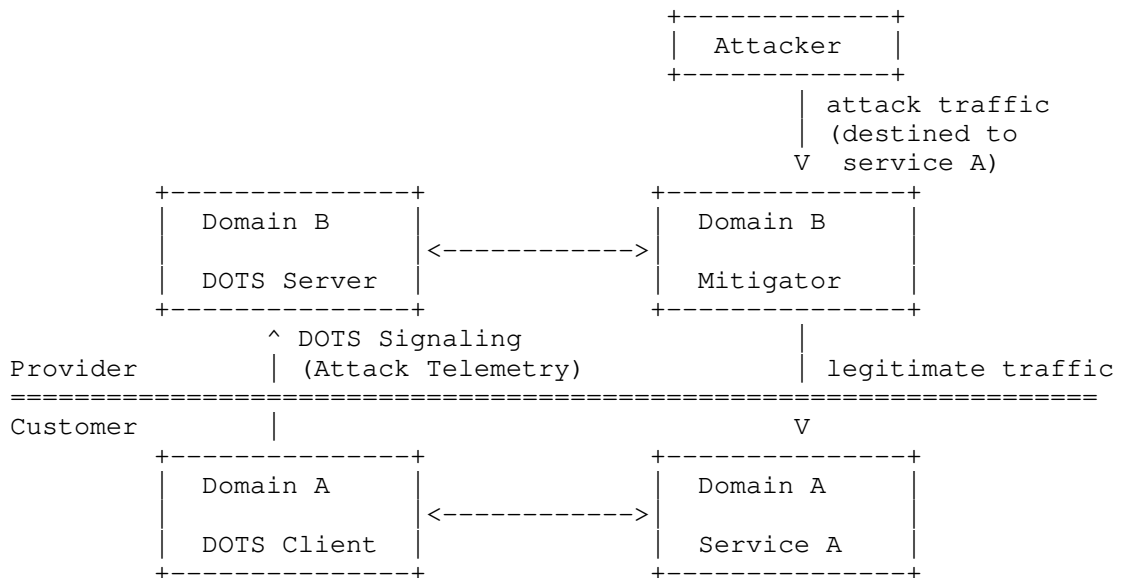


Figure 1: Usecase 1: Single-home Model

## 4.1.2. Usecase 2: Multi-home Model

In the multi-home model, there are one DOTS client and multi DOTS servers. The DOTS client can use both DOTS servers.

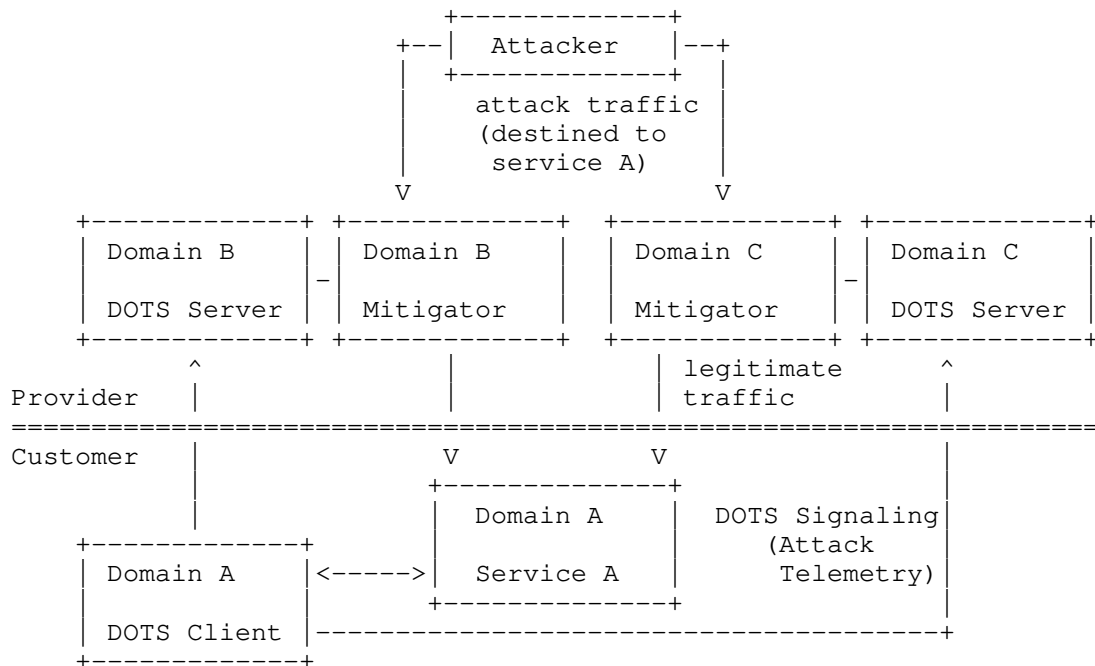


Figure 2: Usecase 2: Multi-home Model

An example of this situation is that an organization is connected to two transit providers. When the customer get attacked, the DDoS traffic would come from transit B and C. Signaling to the DOTS server in transit B can stop only the DDoS traffic from transit B, and vice verse. After detecting the DDoS attack, the DOTS client can send attack telemetry, in which the IP address of the service under attack must be included, to the both DOTS server via DOTS signaling at the same time. Common interface of DOTS signaling will shorten the lead time of the DDoS protection on both transits.

Another example of this situation is cloud type of DDoS mitigation service providers. Cloud type of DDoS mitigation service providers divert traffic to its own domain using DNS or routing protocols, that is BGP route injection. Though they need to provision the returning path mostly on the tunnel interface because they are not directly connected to the domains of the DOTS client, they can accommodate customers remotely.

#### 4.2. Provider-to-Provider Cases

In these cases, a DOTS server in a provider send DOTS request to other providers. If the capacity of the protection system of the provider is insufficient to protect the customer, the task of the protection can be delegated to other DDoS protection providers. The DOTS server in the provider can be a DOTS client of the other DOTS servers in the other providers. The mitigator can delegate the burden of the mitigation, therefore they can accommodate more services which exceed the capacity of its own platform.

##### 4.2.1. Usecase 3: Delegation Model

In the delegation model, a DOTS server is a DOTS client of the other DOTS servers at the same time.

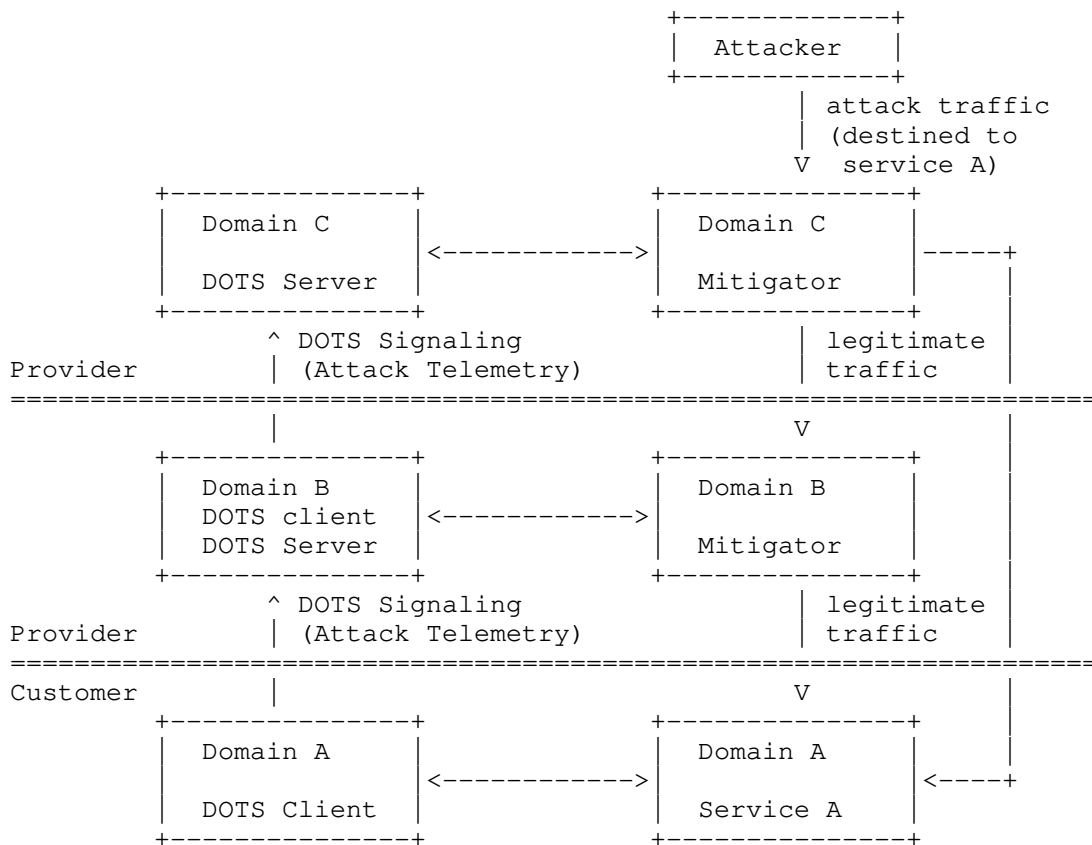


Figure 3: Usecase 3: Delegation Model

If the capacity of the mitigator in provider B is insufficient in comparison with ongoing DDoS attack, the DOTS server in B can be a DOTS client of the DOTS server in C. It needs to be considered whether or not the attack telemetry from A to B (client-to-provider) is the same as the attack telemetry from B to C (provider-to-provider). By just relaying the DOTS signaling information to the DOTS server in domain C, the mitigator in domain C could protect the service A. The DOTS client in A might not notice that the protection was delegated to other domain. However, if the circuit between domain A and domain B is saturated, attack telemetry derived from the observation point of domain A could be insufficient to protect the service. The overwritten attack telemetry derived from observation point of domain B would make the protection more precise. In addition, the returning path of the legitimate traffic also needs to be considered. The mitigator in domain C can return the legitimate traffic to domain B or domain A. In the former case, the attack traffic could re-enter the protection system of the domain B. In the latter case, the returning path information from domain C to domain A might need to be registered in advance. Even if the capacity of the protection system in domain B is enough, in some cases, it is effective to delegate the protection to upstream domain C because stopping DDoS traffic at an ingress border will reduce unnecessary forwarding.

#### 4.2.2. Usecase 4: Distributed Architecture Model

The distributed architecture is one of the multi-provider coordinated DDoS protection, which is a cluster of mutual delegation relations.

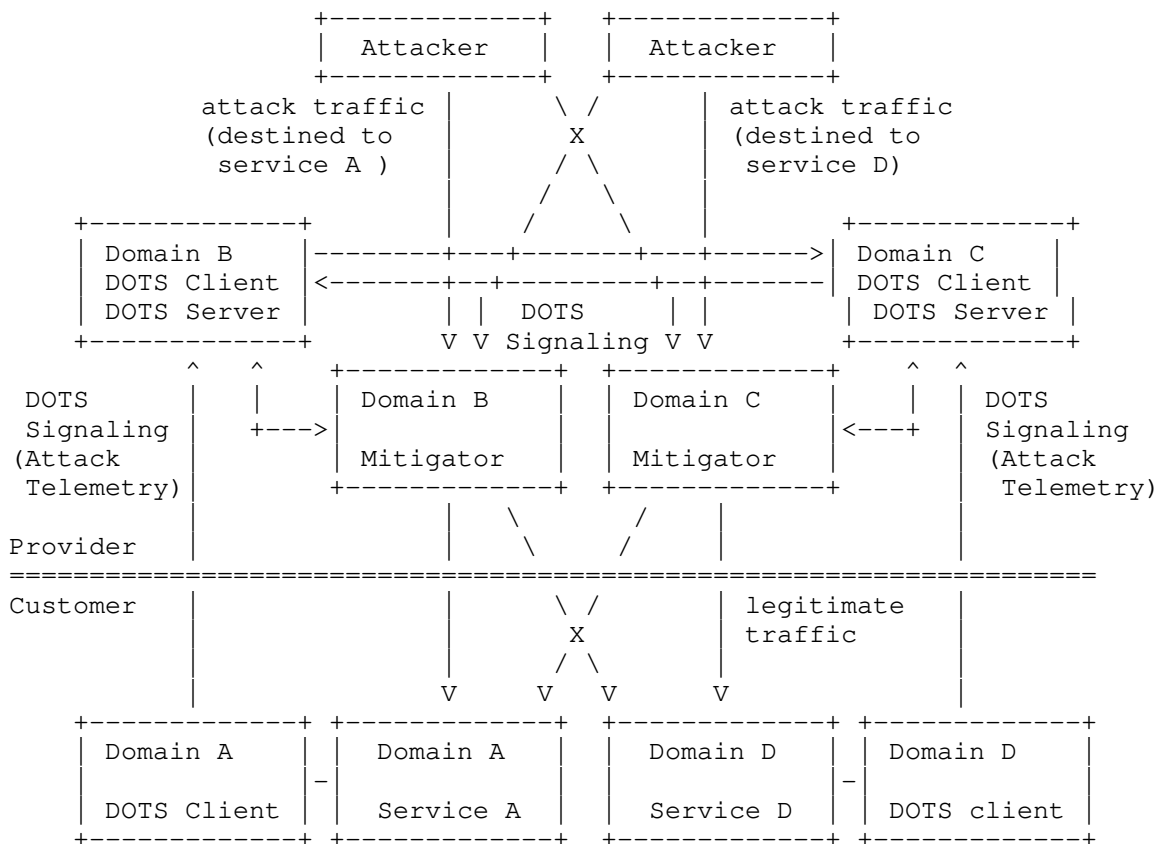


Figure 4: Usecase 4: Mutual Delegation Model

The DOTS client in domain A ask for help to the DOTS server in domain B. Then the DOTS server in domain B delegate the protection to the DOTS server in domain C. The mitigator in domain C protect the service of domain A. On the other hand, the DOTS client in domain D ask for help to the DOTS server in domain C. Then the DOTS server in domain C delegate the protection to the DOTS server in domain B. The mitigator in domain B protect the service of domain D. In this model, the DOTS element in domain B and C is delegating the protection each other. They can leverage total capacity of the mitigator by utilizing the others facility.

If the number of the providers involving the coordinated protection increased and letting them make mutual(peer-to-peer) relationship between each other, that is distributed architecture of cooperative DDoS protection. It becomes difficult to select appropriate DDoS protection according to the capacities of the each mitigator. In

this case, billing data could be more important to adjust the cost distribution fairly.

#### 4.2.3. Usecase 5: Centralized Architecture Model

The centralized architecture model is another multi-provider coordinated DDoS protection, which could overcome the disadvantages of the distributed architecture.

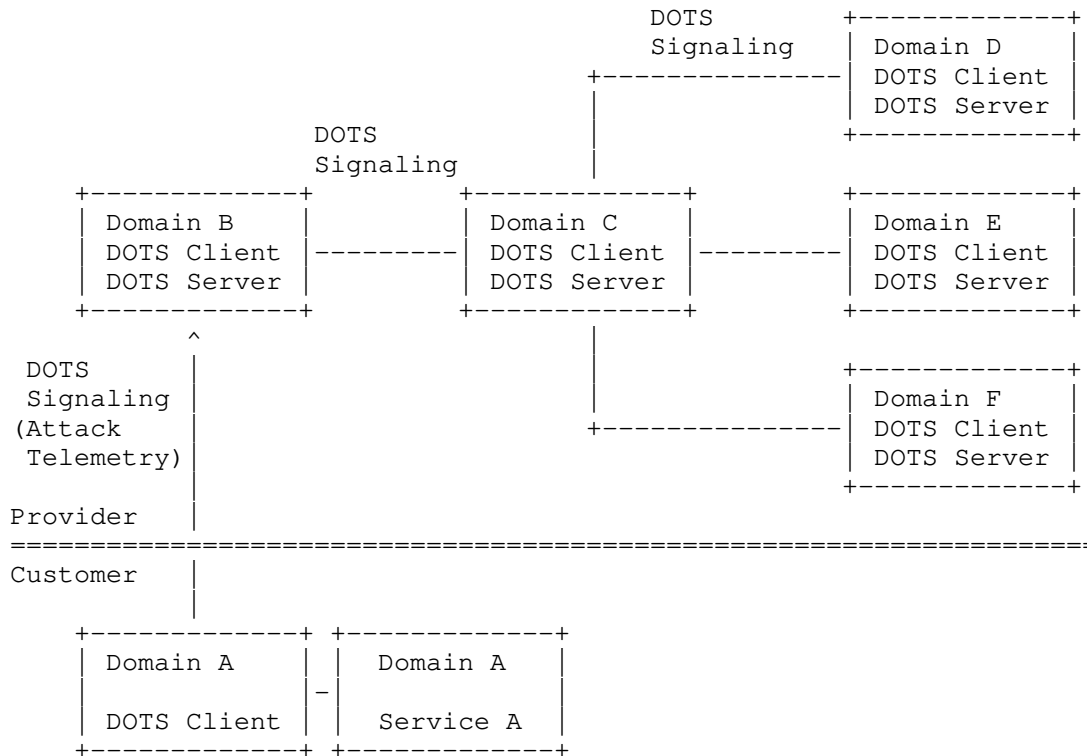


Figure 5: Usecase 5: Centralized Architecture Model

In this model, the DOTS server in domain B can utilize the protection service in domain C, D, E and F. The DOTS server in domain C coordinates the protection services of these providers centrally. The further discussion about the centralized architecture and the distributed architecture is described in [draft-nishizuka-dots-inter-domain-mechanism]

## 5. Security Considerations

As described in the protection methods section, the DOTS framework can be another attack vector to other organizations. Only the legitimate DOTS client should be able to communicate with the DOTS server and the protecting IP address in the request should be checked and restricted in order to eliminate the risks of abuse.

## 6. IANA Considerations

No need to describe any request regarding number assignment.

## 7. References

### 7.1. 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>>.
- [RFC2784] D. Farinacci., T. Li., S. Hanks., D. Meyer., and P. Traina., "Generic Routing Encapsulation (GRE)", March 2000".
- [RFC3882] D. Turk. Bell Canada, "Configuring BGP to Block Denial-of-Service Attacks, September 2004".
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules, August 2009".
- [RFC5635] W. Kumari. and D. McPherson., "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF), August 2009".
- [I-D.draft-ietf-grow-blackholing]  
T. King., C. Dietzel., J. Snijders., G. Doering., and G. Hankins., "BLACKHOLE BGP Community for Blackholing, draft-ietf-grow-blackholing-00, November 2015".
- [I-D.draft-ietf-dots-requirements]  
A. Mortensen., R. Moskowitz., and T. Reddy., "DDoS Open Threat Signaling Requirements, draft-ietf-dots-requirements-00, October 2015".

[I-D.draft-ietf-dots-use-cases]

R. Dobbins, Ed., S. Fouant., D. Migault., R. Moskowitz.,  
N. Teague., L. Xia, "Use cases for DDoS Open Threat  
Signaling, October 2015".

[I-D.draft-reddy-dots-transport]

T. Reddy., D. Wing., P. Patil., M. Geller., M. Boucadair.,  
and R. Moskowitz., "Co-operative DDoS Mitigation, October  
2015".

[I-D.draft-nishizuka-dots-inter-domain-mechanism]

K. Nishizuka., L. Xia., J. Xia., D. Zhang., and L. Fang.,  
"Inter-domain cooperative DDoS protection problems and  
mechanism, February 2016".

## 7.2. URL References

[Cloudflare]

Cloudflare, "<https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/>".

## Author's Address

Kaname Nishizuka  
NTT Communications  
GranPark 16F  
3-4-1 Shibaura, Minato-ku, Tokyo  
108-8118, Japan

E-Mail: kaname@nttv6.jp



DOTS  
Internet-Draft  
Intended status: Standards Track  
Expires: September 17, 2016

T. Reddy  
D. Wing  
P. Patil  
M. Geller  
Cisco  
M. Boucadair  
Orange  
R. Moskowitz  
HTT Consulting  
March 16, 2016

Co-operative DDoS Mitigation  
draft-reddy-dots-transport-03

Abstract

This document discusses mechanisms that a DOTS client can use, when it detects a potential Distributed Denial-of-Service (DDoS) attack, to signal that the DOTS client is under an attack or request an upstream DOTS server to perform inbound filtering in its ingress routers for traffic that the DOTS client wishes to drop. The DOTS server can then undertake appropriate actions (including, blackhole, drop, rate-limit, or add to watch list) on the suspect traffic to the DOTS client, thus reducing the effectiveness of the attack.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 17, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Notational Conventions . . . . .	3
3. Solution Overview . . . . .	3
4. Happy Eyeballs for DOTS Signal Channel . . . . .	6
5. Performance Considerations . . . . .	7
6. DOTS Signal Channel . . . . .	8
6.1. Mitigation Service Request . . . . .	9
6.1.1. Convey DOTS Signals . . . . .	9
6.1.2. Withdraw a DOTS Signal . . . . .	11
6.1.3. Retrieving a DOTS Signal . . . . .	12
6.1.4. Efficacy Update from DOTS Client . . . . .	14
7. DOTS Data Channel . . . . .	14
7.1. Filtering Rules . . . . .	15
7.1.1. Install Filtering Rules . . . . .	15
7.1.2. Remove Filtering Rules . . . . .	17
7.1.3. Retrieving Installed Filtering Rules . . . . .	17
8. IANA Considerations . . . . .	18
9. Security Considerations . . . . .	18
10. Acknowledgements . . . . .	19
11. References . . . . .	19
11.1. Normative References . . . . .	19
11.2. Informative References . . . . .	19
Appendix A. BGP . . . . .	21
Authors' Addresses . . . . .	21

## 1. Introduction

A distributed denial-of-service (DDoS) attack is an attempt to make machines or network resources unavailable to their intended users. In most cases, sufficient scale can be achieved by compromising enough end-hosts and using those infected hosts to perpetrate and amplify the attack. The victim in this attack can be an application server, a client, a router, a firewall, or an entire network, etc.

In a lot of cases, it may not be possible for an enterprise to determine the cause for an attack, but instead just realize that

certain resources seem to be under attack. The document proposes that, in such cases, the DOTS client just inform the DOTS server that the enterprise is under a potential attack and that the DOTS server monitor traffic to the enterprise to mitigate any possible attack. This document also describes a means for an enterprise, which act as DOTS clients, to dynamically inform its DOTS server of the IP addresses or prefixes that are causing DDoS. A DOTS server can use this information to discard flows from such IP addresses reaching the customer network.

The proposed mechanism can also be used between applications from various vendors that are deployed within the same network, some of them are responsible for monitoring and detecting attacks while others are responsible for enforcing policies on appropriate network elements. This cooperations contributes to a ensure a highly automated network that is also robust, reliable and secure. The advantage of the proposed mechanism is that the DOTS server can provide protection to the DOTS client from bandwidth-saturating DDoS traffic.

How a DOTS server determines which network elements should be modified to install appropriate filtering rules is out of scope. A variety of mechanisms and protocols (including NETCONF) may be considered to exchange information through a communication interface between the server and these underlying elements; the selection of appropriate mechanisms and protocols to be invoked for that interfaces is deployment-specific.

Terminology and protocol requirements for co-operative DDoS mitigation are obtained from [I-D.ietf-dots-requirements].

## 2. Notational Conventions

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].

## 3. Solution Overview

Network applications have finite resources like CPU cycles, number of processes or threads they can create and use, maximum number of simultaneous connections it can handle, limited resources of the control plane, etc. When processing network traffic, such an application uses these resources to offer its intended task in the most efficient fashion. However, an attacker may be able to prevent the application from performing its intended task by causing the application to exhaust the finite supply of a specific resource.

TCP DDoS SYN-flood is a memory-exhaustion attack on the victim and ACK-flood is a CPU exhaustion attack on the victim ([RFC4987]). Attacks on the link are carried out by sending enough traffic such that the link becomes excessively congested, and legitimate traffic suffers high packet loss. Stateful firewalls can also be attacked by sending traffic that causes the firewall to hold excessive state and the firewall runs out of memory, and can no longer instantiate the state required to pass legitimate flows. Other possible DDoS attacks are discussed in [RFC4732].

In each of the cases described above, some of the possible arrangements to mitigate the attack are:

- o If a DOTS client determines it is under an attack, the DOTS client can notify the DOTS server using the DOTS signal that it is under a potential attack and request that the DOTS server take precautionary measures to mitigate the attack. The DOTS server can enable mitigation on behalf of the DOTS client by communicating the DOTS client's request to the mitigator and relaying any mitigator feedback to the requesting DOTS client.
- o If a DOTS client determines it is under an attack, the DOTS client can notify its servicing router (DOTS relay) using the DOTS signal that it is under a potential attack and request that the DOTS relay take precautionary measures to mitigate the attack. The DOTS relay propagates the DOTS signal to a DOTS server.

The DOTS server can enable mitigation on behalf of the DOTS relay by communicating the DOTS relay's request to the mitigator and relaying any mitigator feedback to the DOTS relay which in turn propagates the feedback to the requesting DOTS client.

The DOTS client must authenticates itself to the DOTS relay, which in turn authenticates itself to a DOTS server, creating a two-link chain of transitive authentication between the DOTS client and the DOTS server.

- o If a network resource detects a potential DDoS attack from a set of IP addresses, the network resource (DOTS client) informs its servicing router (DOTS relay) of all suspect IP addresses that need to be blocked or black-listed for further investigation.

The DOTS client could also specify a list of protocols and ports in the black-list rule. That DOTS relay in-turn propagates the black-listed IP addresses to the DOTS server and the DOTS server blocks traffic from these IP addresses to the DOTS client thus reducing the effectiveness of the attack.

The DOTS client periodically queries the DOTS server to check the counters mitigating the attack. If the DOTS client receives a response that the counters have not incremented then it can instruct the black-list rules to be removed. If a blacklisted IPv4 address is shared by multiple subscribers, then the side effect of applying the black-list rule will be that traffic from non-attackers will also be blocked by the access network [RFC6269].

A network diagram showing a deployment of these elements is shown below. This shows the DOTS server operating on the access network.

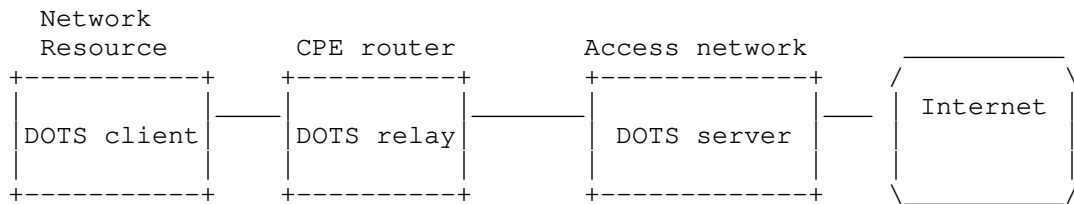


Figure 1

The DOTS server can also be running on the Internet, as depicted below:

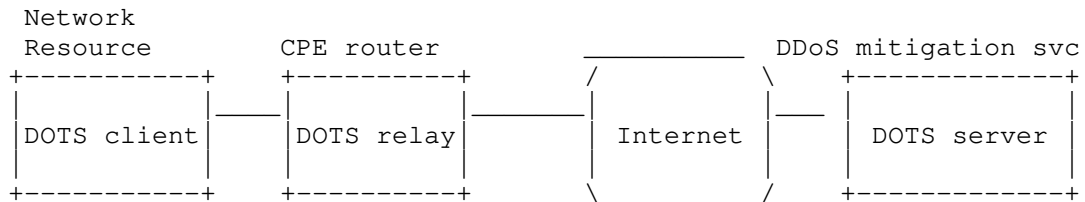


Figure 2

In typical deployments, the DOTS client belongs to a different administrative domain than the DOTS server. For example, the DOTS client is a web server serving content owned and operated by a company, while the DOTS server is owned and operated by a different company providing DDoS mitigation services. That company providing DDoS mitigation service might, or might not, also provide Internet access service to the website operator.

The DOTS server may (not) be co-located with the DOTS mitigator. In typical deployments, the DOTS server belongs to the same administrative domain as the mitigator.

The DOTS client can communicate directly with the DOTS server or indirectly with the DOTS server via the DOTS relay.

#### 4. Happy Eyeballs for DOTS Signal Channel

DOTS signaling can happen with DTLS over UDP and TLS over TCP. A DOTS client can use DNS to determine the IP address(es) of a DOTS server. The DOTS client must know a DOTS server's domain name; hard-coding the domain name of the DOTS server into software is NOT RECOMMENDED in case the domain name is not valid or needs to change for legal or other reasons. The DOTS client performs A and/or AAAA record lookup of the domain name and the result will be a list of IP addresses, each of which can be used to contact the DOTS server using UDP and TCP.

If an IPv4 path to reach a DOTS server is found, but the DOTS server's IPv6 path is not working, a dual-stack DOTS client can experience a significant connection delay compared to an IPv4-only DOTS client. The other problem is that if a middlebox between the DOTS client and DOTS server is configured to block UDP, the DOTS client will fail to establish a DTLS session [RFC6347] with the DOTS server and will, then, have to fall back to TLS over TCP [RFC5246] incurring significant connection delays.

[I-D.ietf-dots-requirements] discusses that DOTS client and server will have to support both connectionless and connection-oriented protocols.

To overcome these connection setup problems, the DOTS client can try connecting to the DOTS server using both IPv6 and IPv4, and try both DTLS over UDP and TLS over TCP in a fashion similar to the Happy Eyeballs mechanism [RFC6555]. These connection attempts are performed by the DOTS client when it initializes, and the client uses that information for its subsequent alert to the DOTS server. In order of preference (most preferred first), it is UDP over IPv6, UDP over IPv4, TCP over IPv6, and finally TCP over IPv4, which adheres to address preference order [RFC6724] and the DOTS preference that UDP be used over TCP (to avoid TCP's head of line blocking).

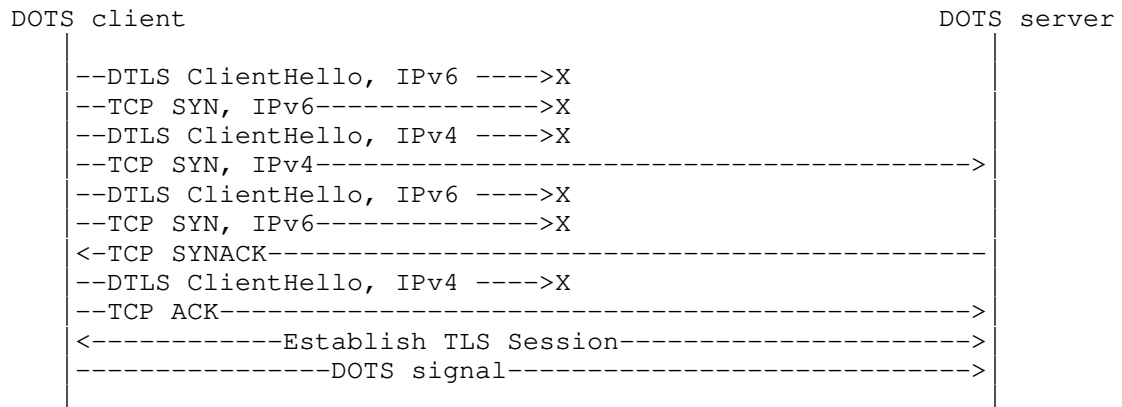


Figure 3: Happy Eyeballs

In reference to Figure 3, the DOTS client sends two TCP SYNs and two DTLS ClientHello messages at the same time over IPv6 and IPv4. In this example, it is assumed that the IPv6 path is broken and UDP is dropped by a middle box but has little impact to the DOTS client because there is no long delay before using IPv4 and TCP. The IPv6 path and UDP over IPv6 and IPv4 is retried until the DOTS client gives up.

## 5. Performance Considerations

DOTS client and server can also use the following techniques to reduce the delay required to deliver a DOTS signal:

- o DOTS client can use (D)TLS session resumption without server-side state [RFC5077] to resume session and convey the DOTS signal.
- o TLS False Start [I-D.ietf-tls-falsestart] which reduces round-trips by allowing the TLS second flight of messages (ChangeCipherSpec) to also contain the DOTS signal.
- o Cached Information Extension [I-D.ietf-tls-cached-info] which avoids transmitting the server's certificate and certificate chain if the client has cached that information from a previous TLS handshake.
- o TCP Fast Open [RFC7413] can reduce the number of round-trips to convey DOTS signal.
- o While the communication to the DOTS server is quiescent, the DOTS client may want to probe the server to ensure it has maintained cryptographic state. Such probes can also keep alive firewall or

NAT bindings. This probing reduces the frequency of needing a new handshake when a DOTS signal needs to be conveyed to the DOTS server.

- \* A DTLS heartbeat [RFC6520] verifies the DOTS server still has DTLS state by returning a DTLS message. If the server has lost state, it returns a DTLS Alert. Upon receipt of an unauthenticated DTLS Alert, the DTLS client validates the Alert is within the replay window (Section 4.1.2.6 of [RFC6347]). It is difficult for the DTLS client to validate the DTLS Alert was generated by the DTLS server in response to a request or was generated by an on- or off-path attacker. Thus, upon receipt of an in-window DTLS Alert, the client SHOULD continue re-transmitting the DTLS packet (in the event the Alert was spoofed), and at the same time it SHOULD initiate DTLS session resumption.
- \* TLS runs over TCP, so a simple probe is a 0-length TCP packet (a "window probe"). This verifies the TCP connection is still working, which is also sufficient to prove the server has retained TLS state, because if the server loses TLS state it abandons the TCP connection. If the server has lost state, a TCP RST is returned immediately.

## 6. DOTS Signal Channel

A DOTS client can use RESTful APIs discussed in this section to signal/inform a DOTS server of an attack.

TBD: Constrained Application Protocol (CoAP) [RFC7252] is used for DOTS signal channel. CoAP was designed according to the REST architecture, and thus exhibits functionality similar to that of the HTTP protocol, it is quite straightforward to map from CoAP to HTTP and from HTTP to CoAP. CoAP has been defined to make use of both DTLS over UDP and TLS over TCP. The advantages of CoAP are: (1) Like HTTP, CoAP is based on the successful REST model, (2) CoAP is designed to use minimal resources, (3) CoAP integrates with JSON, CBOR or any other data format, (4) asynchronous message exchanges, etc.

JSON [RFC7159] payloads is be used to convey signal channel specific payload messages that convey request parameters and response information such as errors.



### 6.1. Mitigation Service Request

The following APIs define the means to convey a DOTS signal from a DOTS client to a DOTS server. POST request is used to convey the DOTS signal from a DOTS client to a DOTS server over the signal channel, possibly traversing a DOTS relay, indicating the DOTS client's need for mitigation, as well as the scope of any requested mitigation (Section 6.1.1).

DELETE requests are used by the DOTS client to withdraw the request for mitigation from the DOTS server (Section 6.1.2).

GET requests are used by the DOTS client to retrieve the DOTS signal(s) it had conveyed to the DOTS server (Section 6.1.3).

PUT requests are used by the DOTS client to convey mitigation efficacy updates to the DOTS server (Section 6.1.4).

#### 6.1.1. Convey DOTS Signals

An HTTP POST request is used to convey a DOTS signal to the DOTS server (Figure 4).

```
POST {scheme}://{host}:{port}/.well-known/{version}/{URI suffix for DOTS signal}
Accept: application/json
Content-type: application/json
{
  "policy-id": "number",
  "target-ip": "string",
  "target-port": "string",
  "target-protocol": "string",
  "lifetime": "number"
}
```

Figure 4: POST to convey DOTS signals

The header fields are described below.

**policy-id:** Identifier of the policy represented using a number. This identifier MUST be unique for each policy bound to the DOTS client, i.e., the policy-id needs to be unique relative to the active policies with the DOTS server. This identifier must be generated by the DOTS client. This document does not make any assumption about how this identifier is generated. This is a mandatory attribute.

**target-ip:** A list of IP addresses or prefixes under attack. This is an optional attribute.

**target-port:** A list of ports under attack. This is an optional attribute.

**target-protocol:** A list of protocols under attack. Valid protocol values include tcp, udp, sctp, and dccp. This is an optional attribute.

**lifetime:** Lifetime of the mitigation request policy in seconds. Upon the expiry of this lifetime, and if the request is not refreshed, the mitigation request is removed. The request can be refreshed by sending the same message again. The default lifetime of the policy is 60 minutes -- this value was chosen to be long enough so that refreshing is not typically a burden on the DOTS client, while expiring the policy where the client has unexpectedly quit in a timely manner. A lifetime of zero indicates indefinite lifetime for the mitigation request. The server **MUST** always indicate the actual lifetime in the response. This is an optional attribute in the request.

The relative order of two rules is determined by comparing their respective policy identifiers. The rule with lower numeric policy identifier value has higher precedence (and thus will match before) than the rule with higher numeric policy identifier value.

To avoid DOTS signal message fragmentation and the consequently decreased probability of message delivery, DOTS agents **MUST** ensure that the DTLS record **MUST** fit within a single datagram. If the Path MTU is not known to the DOTS server, an IP MTU of 1280 bytes **SHOULD** be assumed. The length of the URL **MUST NOT** exceed 256 bytes. If UDP is used to convey the DOTS signal and the request size exceeds the Path MTU then the DOTS client **MUST** split the DOTS signal into separate messages, for example the list of addresses in the 'target-ip' field could be split into multiple lists and each list conveyed in a new POST request.

Implementation Note: DOTS choice of message size parameters works well with IPv6 and with most of today's IPv4 paths. However, with IPv4, it is harder to absolutely ensure that there is no IP fragmentation. If IPv4 support on unusual networks is a consideration and path MTU is unknown, implementations may want to limit themselves to more conservative IPv4 datagram sizes such as 576 bytes, as per [RFC0791] IP packets up to 576 bytes should never need to be fragmented, thus sending a maximum of 500 bytes of DOTS signal over a UDP datagram will generally avoid IP fragmentation.

Figure 5 shows a POST request to signal that ports 80, 8080, and 443 on the servers 2002:db8:6401::1 and 2002:db8:6401::2 are being attacked.

```
POST https://www.example.com/.well-known/v1/DOTS signal
Accept: application/json
Content-type: application/json
{
  "policy-id":123321333242,
  "target-ip":[
    "2002:db8:6401::1",
    "2002:db8:6401::2"
  ],
  "target-port":[
    "80",
    "8080",
    "443"
  ],
  "target-protocol":"tcp"
}
```

Figure 5: POST for DOTS signal

The DOTS server indicates the result of processing the POST request using HTTP response codes. HTTP 2xx codes are success, HTTP 4xx codes are some sort of invalid request and HTTP 5xx codes are returned if the DOTS server has erred or is incapable of performing the mitigation. Response code 200 (OK) will be returned in the response if the DOTS server has accepted the mitigation request and will try to mitigate the attack. If the request is missing one or more mandatory attributes then 400 (Bad Request) will be returned in the response or if the request contains invalid or unknown parameters then 400 (Invalid query) will be returned in the response. The HTTP response will include the JSON body received in the request.

#### 6.1.2. Withdraw a DOTS Signal

An HTTP DELETE request is used to withdraw a DOTS signal from a DOTS server (Figure 6).

```
DELETE {scheme}://{host}:{port}/.well-known/{URI suffix for DOTS signal}
Accept: application/json
Content-type: application/json
{
  "policy-id": "number"
}
```

Figure 6: Withdraw DOTS signal

If the DOTS server does not find the policy number conveyed in the DELETE request in its policy state data, then it responds with "404" HTTP error response code. The DOTS server successfully acknowledges

a DOTS client's request to withdraw the DOTS signal using 200 (OK) response code, and ceases mitigation activity as quickly as possible.

#### 6.1.3. Retrieving a DOTS Signal

An HTTP GET request is used to retrieve information and status of a DOTS signal from a DOTS server (Figure 7). If the DOTS server does not find the policy number conveyed in the GET request in its policy state data then it responds with a 404 HTTP error response code.

- 1) To retrieve all DOTS signals signaled by the DOTS client.

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix for DOTS signal}/list
```

- 2) To retrieve a specific DOTS signal signaled by the DOTS client.  
The policy information in the response will be formatted in the same order it was processed at the DOTS server.

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix for DOTS signal}/<policy-id  
number>
```

Figure 7: GET to retrieve the rules

Figure 8 shows the response of all the active policies on the DOTS server.

```
{
  "policy-data":[
    {
      "policy-id":123321333242,
      "target-prtoocol":"tcp",
      "lifetime":3600,
      "status":"mitigation in progress"
    },
    {
      "policy-id":123321333244,
      "target-protocol":"udp",
      "lifetime":1800,
      "status":"mitigation complete"
    },
    {
      "policy-id":123321333245,
      "target-protocol":"tcp",
      "lifetime":1800,
      "status":"attack stopped"
    }
  ]
}
```

Figure 8: Response body

The various possible values of status field are explained below:

mitigation in progress: Attack mitigation is in progress (for e.g., changing the network path to re-route the inbound traffic to DOTS mitigator).

mitigation complete: Attack is successfully mitigated (for e.g., attack traffic is dropped).

attack stopped: Attack has stopped and the DOTS client can withdraw the mitigation request.

#### 6.1.3.1. Mitigation Status

A DOTS client retrieves the information about a DOTS signal at frequent intervals to determine the status of an attack. If the DOTS server has been able to mitigate the attack and the attack has stopped, the DOTS server indicates as such in the status, and the DOTS client recalls the mitigation request.

A DOTS client should react to the status of the attack from the DOTS server and not the fact that it has recognized, using its own means, that the attack has been mitigated. This ensures that the DOTS

client does not recall a mitigation request in a premature fashion because it is possible that the DOTS client does not sense the DDOS attack on its resources but the DOTS server could be actively mitigating the attack and the attack is not completely averted.

#### 6.1.4. Efficacy Update from DOTS Client

While DDoS mitigation is active, a DOTS client MAY frequently transmit DOTS mitigation efficacy updates to the relevant DOTS server. An HTTP PUT request (Figure 9) is used to convey the mitigation efficacy update to the DOTS server. The PUT request MUST include all the header fields used in the POST request to convey the DOTS signal (Section 6.1.1). If the DOTS server does not find the policy number conveyed in the PUT request in its policy state data, it responds with a 404 HTTP error response code.

```
PUT {scheme}://{host}:{port}/.well-known/{URI suffix for DOTS signal}/<policy-  
id number>  
Accept: application/json  
Content-type: application/json  
{  
  "target-ip": "string",  
  "target-port": "string",  
  "target-protocol": "string",  
  "lifetime": "number",  
  "attack-status": "string"  
}
```

Figure 9: Efficacy Update

The 'attack-status' field is a mandatory attribute. The various possible values contained in the 'attack-status' field are explained below:

in-progress: DOTS client determines that it is still under attack.

terminated: Attack is successfully mitigated (e.g., attack traffic is dropped).

## 7. DOTS Data Channel

A DOTS client can use RESTful APIs to provision and manage filters on the DOTS server. TBD: The data channel is intended to be used for bulk data exchanges and requires a reliable transport, CoAP over TLS over TCP is used for data channel.

JSON [RFC7159] payloads is used to convey both filtering rules as well as data channel specific payload messages that convey request parameters and response information such as errors. All data channel

URIs defined in this document, and in subsequent documents, MUST NOT have a URI containing "/DOTS signal".

One of the possible arrangements for DOTS client to signal filtering rules to the DOTS server via the DOTS relay is discussed below:

The DOTS conveys the black-list rules to the DOTS relay. The DOTS relay validates if the DOTS client is authorized to signal the black-list rules and if the client is authorized propagates the rules to the DOTS server. Likewise, the DOTS server validates if the DOTS relay is authorized to signal the black-list rules. To create or purge filters, the DOTS client sends HTTP requests to the DOTS relay. The DOTS relay acts as an proxy, validates the rules and proxies the requests containing the black-listed IP addresses to the DOTS server. When the DOTS relay receives the associated HTTP response from the DOTS server, it propagates the response back to the DOTS client. If an attack is detected by the DOTS relay then it can act as a DOTS client and signal the black-list rules to the DOTS server. The DOTS relay plays the role of both client and proxy.

### 7.1. Filtering Rules

The following APIs define means for a DOTS client to configure filtering rules on a DOTS server.

#### 7.1.1. Install Filtering Rules

An HTTP POST request is used to push filtering rules to a DOTS server (Figure 10).

```
POST {scheme}://{host}:{port}/.well-known/{version}/{URI suffix for filtering}
Accept: application/json
Content-type: application/json
{
  "policy-id": "number",
  "traffic-protocol": "string",
  "source-protocol-port": "string",
  "destination-protocol-port": "string",
  "destination-ip": "string",
  "source-ip": "string",
  "lifetime": "number",
  "traffic-rate" : "number"
}
```

Figure 10: POST to install filtering rules

The header fields are described below:

**policy-id:** Identifier of the policy represented using a number.

This identifier MUST be unique for each policy bound to the DOTS client, i.e., the policy-id needs to be unique relative to the active policies with the DOTS server. This identifier must be generated by the client. This document does not make any assumption about how this identifier is generated. This is an mandatory attribute.

**traffic-protocol:** Valid protocol values include tcp, udp, sctp, and dccp. This is an mandatory attribute.

**source-protocol-port:** The source port number, port number range (using "-"). For TCP, UDP, SCTP, or DCCP: the source range of ports (e.g., 1024-65535). This is an optional attribute.

**destination-protocol-port:** The destination port number, port number range (using "-"). For TCP, UDP, SCTP, or DCCP: the destination range of ports (e.g., 443-443). This information is useful to avoid disturbing a group of customers when address sharing is in use [RFC6269]. This is an optional attribute.

**destination-ip:** The destination IP address, IP addresses separated by commas, or prefixes using "/" notation. This is an optional attribute.

**source-ip:** The source IP addresses, IP addresses separated by commas, or prefixes using "/" notation. This is an optional attribute.

**lifetime:** Lifetime of the rule in seconds. Upon the expiry of this lifetime, and if the request is not refreshed, this particular rule is removed. The rule can be refreshed by sending the same message again. The default lifetime of the rule is 60 minutes -- this value was chosen to be long enough so that refreshing is not typically a burden on the DOTS client, while expiring the rule where the client has unexpectedly quit in a timely manner. A lifetime of zero indicates indefinite lifetime for the rule. The server MUST always indicate the actual lifetime in the response. This is an optional attribute in the request.

**traffic-rate:** This is the allowed traffic rate in bytes per second indicated in IEEE floating point [IEEE.754.1985] format. The value 0 indicates all traffic for the particular flow to be discarded. This is a mandatory attribute.

The relative order of two rules is determined by comparing their respective policy identifiers. The rule with lower numeric policy



identifier value has higher precedence (and thus will match before) than the rule with higher numeric policy identifier value.

Figure 11 shows a POST request to block traffic from attacker IPv6 prefix 2001:db8:abcd:3f01::/64 to network resource using IPv6 address 2002:db8:6401::1 to operate a server on TCP port 443.

```
POST https://www.example.com/.well-known/v1/filter
Accept: application/json
Content-type: application/json
{
  "policy-id": 123321333242,
  "traffic-protocol": "tcp",
  "source-protocol-port": "0-65535",
  "destination-protocol-port": "443",
  "destination-ip": "2001:db8:abcd:3f01::/64",
  "source-ip": "2002:db8:6401::1",
  "lifetime": 1800,
  "traffic-rate": 0
}
```

Figure 11: POST to Install Black-list Rules

#### 7.1.2. Remove Filtering Rules

An HTTP DELETE request is used to delete filtering rules from a DOTS server (Figure 12).

```
DELETE {scheme}://{host}:{port}/.well-known/{URI suffix for filtering}
Accept: application/json
Content-type: application/json
{
  "policy-id": "number"
}
```

Figure 12: DELETE to remove the rules

#### 7.1.3. Retrieving Installed Filtering Rules

An HTTP GET request is used to retrieve filtering rules from a DOTS server.

Figure 13 shows an example to retrieve all the black-lists rules programmed by the DOTS client while Figure 14 shows an example to retrieve specific black-list rules programmed by the DOTS client.

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix for filtering}
```

Figure 13: GET to retrieve the rules (1)

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix for filtering}
Accept: application/json
Content-type: application/json
{
  "policy-id": "number"
}
```

Figure 14: GET to retrieve the rules (2)

TODO: show response

## 8. IANA Considerations

TODO

## 9. Security Considerations

Authenticated encryption MUST be used for data confidentiality and message integrity. (D)TLS based on client certificate MUST be used for mutual authentication. The interaction between the DOTS agents requires Datagram Transport Layer Security (DTLS) and Transport Layer Security (TLS) with a ciphersuite offering confidentiality protection and the guidance given in [RFC7525] MUST be followed to avoid attacks on (D)TLS.

If TCP is used between DOTS agents, attacker may be able to inject RST packets, bogus application segments, etc., regardless of whether TLS authentication is used. Because the application data is TLS protected, this will not result in the application receiving bogus data, but it will constitute a DoS on the connection. This attack can be countered by using TCP-AO [RFC5925]. If TCP-AO is used, then any bogus packets injected by an attacker will be rejected by the TCP-AO integrity check and therefore will never reach the TLS layer.

Special care should be taken in order to ensure that the activation of the proposed mechanism won't have an impact on the stability of the network (including connectivity and services delivered over that network).

Involved functional elements in the cooperation system must establish exchange instructions and notification over a secure and authenticated channel. Adequate filters can be enforced to avoid that nodes outside a trusted domain can inject request such as deleting filtering rules. Nevertheless, attacks can be initiated

from within the trusted domain if an entity has been corrupted. Adequate means to monitor trusted nodes should also be enabled.

## 10. Acknowledgements

Thanks to Christian Jacquenet, Roland Dobbins, Andrew Mortensen, Roman D. Danyliw, and Gilbert Clark for the discussion and comments.

## 11. References

### 11.1. 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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<http://www.rfc-editor.org/info/rfc5925>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<http://www.rfc-editor.org/info/rfc7252>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<http://www.rfc-editor.org/info/rfc7525>>.

### 11.2. 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-tls-cached-info]  
Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", draft-ietf-tls-cached-info-22 (work in progress), January 2016.
- [I-D.ietf-tls-falsestart]  
Langley, A., Modadugu, N., and B. Moeller, "Transport Layer Security (TLS) False Start", draft-ietf-tls-falsestart-01 (work in progress), November 2015.
- [IEEE.754.1985]  
Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", August 1985.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [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>>.
- [RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, DOI 10.17487/RFC4987, August 2007, <<http://www.rfc-editor.org/info/rfc4987>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <<http://www.rfc-editor.org/info/rfc5077>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<http://www.rfc-editor.org/info/rfc5575>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<http://www.rfc-editor.org/info/rfc6269>>.
- [RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <<http://www.rfc-editor.org/info/rfc6520>>.

- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <<http://www.rfc-editor.org/info/rfc6555>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<http://www.rfc-editor.org/info/rfc6724>>.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<http://www.rfc-editor.org/info/rfc7159>>.
- [RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <<http://www.rfc-editor.org/info/rfc7413>>.

#### Appendix A. BGP

BGP defines a mechanism as described in [RFC5575] that can be used to automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate DDoS attacks. However, support for BGP in an access network does not guarantee that traffic filtering will always be honored. Since a DOTS client will not receive an acknowledgment for the filtering request, the DOTS client should monitor and apply similar rules in its own network in cases where the DOTS server is unable to enforce the filtering rules. In addition, enforcement of filtering rules of BGP on Internet routers are usually governed by the maximum number of data elements the routers can hold as well as the number of events they are able to process in a given unit of time.

#### Authors' Addresses

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

Email: [tiredy@cisco.com](mailto:tiredy@cisco.com)

Dan Wing  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, California 95134  
USA

Email: [dwing@cisco.com](mailto:dwing@cisco.com)

Prashanth Patil  
Cisco Systems, Inc.

Email: [praspati@cisco.com](mailto:praspati@cisco.com)

Mike Geller  
Cisco Systems, Inc.  
3250  
Florida 33309  
USA

Email: [mgeller@cisco.com](mailto:mgeller@cisco.com)

Mohamed Boucadair  
Orange  
Rennes 35000  
France

Email: [mohamed.boucadair@orange.com](mailto:mohamed.boucadair@orange.com)

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 42837  
United States

Email: [rgm@htt-consult.com](mailto:rgm@htt-consult.com)