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

A. Mortensen
Arbor Networks, Inc.
R. Moskowitz
HTT Consulting
T. Reddy
Cisco Systems, Inc.
October 19, 2015

DDoS Open Threat Signaling Requirements
draft-ietf-dots-requirements-00

Abstract

This document defines the requirements for the DDoS Open Threat Signaling (DOTS) protocols coordinating attack response against Distributed Denial of Service (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 April 21, 2016.

Table of Contents

1. Introduction

1.1. Overview

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, thanks to 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 higher profile and 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

response may be fragmentary or otherwise incomplete, leaving key players in the attack path unable to assist in the defense.

There are many ways to respond to an ongoing DDoS attack, some of them better than others, but the lack of a common method to coordinate a real-time response across layers and network domains inhibits the speed and effectiveness of DDoS attack mitigation.

DOTS was formed to address this lack. The DOTS protocols are therefore not concerned with the form of response, but rather with communicating the need for a response, supplementing the call for help with pertinent details about the detected attack. To achieve this aim, the protocol must permit the DOTS client to request or withdraw a request for coordinated mitigation; to set the scope of mitigation, restricted to the client's network space; and to supply summarized attack information and additional hints the DOTS server elements can use to increase the accuracy and speed of the attack response.

The protocol must also continue to operate even in extreme network conditions. It must be resilient enough to ensure a high probability of signal delivery in spite of high packet loss rates. As such, elements should be in regular, bidirectional contact to measure peer health, provide mitigation-related feedback, and allow for active mitigation adjustments.

Lastly, the protocol must take care to ensure the confidentiality, integrity and authenticity of messages passed between peers to prevent the protocol from being repurposed to contribute to the very attacks it's meant to deflect.

Drawing on the DOTS use cases [I-D.ietf-dots-use-cases] for reference, this document details the requirements for protocols achieving the DOTS goal of standards-based open threat signaling.

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].

The following terms are used to define relationships between elements, the data they exchange, and methods of communication among them:

attack telemetry: collected network traffic characteristics defining the nature of a DDoS attack.

mitigation: A defensive response against a detected DDoS attack, performed by an entity in the network path between attack sources and the attack target, either through inline deployment or some form of traffic diversion. The form mitigation takes is out of scope for this document.

mitigator: A network element capable of performing mitigation of a detected DDoS attack.

DOTS client: A DOTS-aware network element requesting attack response coordination with another DOTS-aware element, with the expectation that the remote element is capable of helping fend off the attack against the client.

DOTS server: A DOTS-aware network element handling and responding to messages from a DOTS client. 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 client. A DOTS server may also be a mitigator.

DOTS relay: A DOTS-aware network element positioned between a DOTS server and a DOTS client. 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.

DOTS agents: A collective term for DOTS clients, servers and 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 keep-alive message transmitted between DOTS agents over the signal channel, used to measure peer health. Heartbeat functionality is not required to be distinct from signal.

client signal: A message sent 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, optionally including detected attack telemetry to supplement server-initiated mitigation.

server signal: A message sent from a DOTS server to a DOTS client 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 the DOTS client, containing information about the status of any requested mitigation and its efficacy.

data channel: A secure communication layer between client and server used for infrequent bulk exchange of data not easily or appropriately communicated through the signal channel under attack conditions.

blacklist: a list of source addresses or prefixes from which traffic should be blocked.

whitelist: a list of source addresses or prefixes from which traffic should always be allowed, regardless of contradictory data gleaned in a detected attack.

2. Requirements

This section describes the required features and characteristics of the DOTS protocols. The requirements are informed by the use cases described in [I-D.ietf-dots-use-cases].

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, by signaling inter- or intra-domain using the DOTS protocol. DOTS clients should similarly be able to withdraw aid requests arbitrarily. Regular feedback between DOTS client and server supplement the defensive alliance by maintaining a common understanding of DOTS peer health and activity. Bidirectional communication between DOTS client and server is therefore critical.

Yet the DOTS protocol 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) with a given network will tend to contribute to the robustness demanded by a viable DOTS protocol.

On the other hand, DOTS must have adequate message confidentiality, integrity and authenticity to keep the protocol from becoming another vector for the very attacks it's meant to help fight off. The DOTS client must be authenticated to the DOTS server, 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

G-001 Interoperability: DOTS's objective is to develop a standard mechanism for signaling detected ongoing DDoS attacks. That objective is unattainable without well-defined specifications for any protocols or data models emerging from DOTS. All protocols, data models and interfaces MUST be detailed enough to ensure interoperable implementations.

G-002 Extensibility: Any protocols or data models developed as part of DOTS MUST be designed to support future extensions. Provided they do not undermine the interoperability and backward compatibility requirements, extensions are a critical part of keeping DOTS adaptable to changing operational and proprietary needs to keep pace with evolving DDoS attack methods.

G-003 Resilience: The signaling protocol MUST be designed to maximize the probability of signal delivery even under the severely constrained network conditions imposed by the attack traffic. The protocol SHOULD be resilient, that is, continue operating despite message loss and out-of-order or redundant signal delivery.

G-004 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.

G-005 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.

G-006 Message Integrity: 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, DOTS agents MUST be able to negotiate the terms and mechanisms of protocol security, subject to the interoperability and signal message size requirements above.

G-007 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, such as including a timestamp or sequence number in every heartbeat and signal sent between DOTS agents.

G-008 Bulk Data Exchange: Infrequent bulk data exchange between DOTS client and server can also significantly augment attack response coordination, permitting such tasks as population of black- or white-listed source addresses; address group aliasing; exchange of incident reports; and other hinting or configuration supplementing attack response.

As the resilience requirements for DOTS mandate small signal message size, a separate, secure data channel utilizing an established reliable protocol SHOULD be used for bulk data exchange. The mechanism for bulk data exchange is not yet specified, but the nature of the data involved suggests use of a reliable, adaptable protocol with established and configurable conventions for authentication and authorization.

2.2. Operational requirements

OP-001 Use of Common Transports: DOTS MUST operate over common standardized transport protocols. While the protocol resilience requirement strongly RECOMMENDS the use of connectionless protocols, in particular the User Datagram Protocol (UDP) [RFC0768], use of a standardized, connection-oriented protocol like the Transmission Control Protocol (TCP) [RFC0793] MAY be necessary due to network policy or middleware limitations.

OP-002 Peer Mutual Authentication: The client and server MUST authenticate each other before a DOTS session is considered

active. 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.

OP-003 Session Health Monitoring: The client and server 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 client or server explicitly ends the session, or either DOTS agent fails to receive heartbeats from the other after a mutually negotiated timeout period has elapsed.

OP-004 Mitigation Capability Opacity: DOTS is a threat signaling protocol. The server and mitigator MUST NOT make any assumption about the attack detection, classification, or mitigation capabilities of the client. While the server and mitigator MAY take hints from any attack telemetry included in client signals, the server and mitigator cannot depend on the client for authoritative attack classification. Similarly, the mitigator cannot assume the client can or will mitigate attack traffic on its own.

The client likewise MUST NOT make any assumptions about the capabilities of the server or mitigator with respect to detection, classification, and mitigation of DDoS attacks. The form of any attack response undertaken by the mitigator is not in scope.

OP-005 Mitigation Status: 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 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-006 Mitigation Scope: DOTS clients MUST indicate the desired address space coverage of any mitigation, for example by using Classless Internet Domain Routing (CIDR) [RFC1518],[RFC1519] prefixes, [RFC2373] for IPv6 prefixes, the length/prefix convention established in the Border Gateway Protocol (BGP) [RFC4271], 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, clients SHOULD include that information in client signals.

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.

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 should be adaptable and 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 may be transactional, requiring reliable, in-order packet delivery.

DATA-002 Data privacy and integrity: Transmissions over the data channel may contain 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 Security Considerations below.) Consequently data sent over the data channel MUST be encrypted and authenticated using current industry best practices.

DATA-003 Mutual authentication: DOTS agents MUST mutually authenticate each other before data may be exchanged over the data channel. DOTS agents MAY take additional steps to authorize data exchange, as in the prefix group example above, before accepting data over the data channel. The form of authentication and authorization is unspecified.

DATA-004 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. Data model requirements

TODO

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: 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.

5. Change Log

5.1. 00 revision

2015-10-15

5.2. Initial revision

2015-09-24 Andrew Mortensen

6. References

6.1. 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>.

6.2. Informative References

[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>.

[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>.

Authors' Addresses

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

Email: amortensen@arbor.net

Robert Moskowitz
HTT Consulting
Oak Park, MI 42837
United States

Email: rgm@htt-consult.com

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

Email: tireddy@cisco.com