DOTS
Internet-Draft
Intended status: Standards Track
Expires: November 4, 2017

J. Francois
Inria
A. Lahmadi
University of Lorraine - LORIA
M. Davids
G. Moura
SIDN Labs

May 3, 2017

IPv6 DOTS Signal Option
draft-francois-dots-ipv6-signal-option-02

Abstract

DOTS client signal using original signal communication channel can expect service degradation and even service disruption as any other service over Internet but in more severe conditions because the signal may have to be transmitted over congested paths due to the denial-of-service attack.

This document specifies a fall-back asynchronous mechanism using an intermediate agent to store DOTS signal information during a limited period of time. This mechanism allows a DOTS server to request a signal information stored by a DOTS client when no heartbeat is received from the DOTS client. This intermediate agent called DOTS Signal Repository have to be connected to the DOTS client and server independently. The repository must be located and/or reached through one or multiple network paths, preferably as most as possible disjoint from regular signal channel, in order to increase its reachability. The document introduces a set of support protocols to build the asynchronous communication between the DOTS cient, server and the repository.

This Internet-Draft will expire on November 4, 2017.

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

A distributed denial-of-service (DDoS) attack aims at rendering machines or network resources unavailable. These attacks have grown in frequency, intensity and target diversity [I-D.ietf-dots-requirements]. Moreover, several protocols have been utilized to amplify the intensity of the attacks [kuhrer2014exit], peaking at several hundred gigabits per second.

DDoS Open Threat Signaling (DOTS) aims at defining a common and open protocol to signal DDoS attacks to facilitate a coordinated response to these attacks. This document specifies an asynchronous signaling method that MAY be used between a DOTS client and server instead of relying on purely synchronous communication as specified in [I-D.ietf-dots-signal-channel]. Indeed initial signaling should be done in real-time through connections between DOTS clients and severs such that a client can forward signal information as soon as the attack is detected. However, the signaling messages may have to be forwarded through paths impacted by the attack itself, i.e. highly congested. Asynchronous signaling in this document is an additional mechanism which MAY propagate signal information in a less reactive manner due to the use of an asynchronous communication channel but through alternative paths in the network. It increases the reachability of the DOTS server which will then in charge of requesting the mitigation.

The proposed mechanism constitutes an additional signaling channel but MUST NOT replace the original signaling channel used between DOTS client and servers as the one defined in [I-D.ietf-dots-signal-channel].

To perform asynchronous communication, this document introduces DOTS Signals Repository (DSR) which represents datastores where DOTS clients can send signal information. This information is then stored and the DOTS server can request it. In addition to provide a general process for achieving asynchronous signaling, this document introduces also a set of protocols which can support it.

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 terms DOTS client, DOTS server, DOTS gateway, DOTS agents refers to the terminology introduced in [I-D.ietf-dots-architecture].

The following terms are introduced:

- DSR (DOTS signal repository): intermediate agent able to store signal from clients during a limited period of time and which can be requested by DOTS servers.

- Opportunistic DOTS signal: an IPv6 packet containing the signaling attributes of an attack within the Hop-by-Hop extension header. The purpose is the same as the DOTS signal. It is used to request help for mitigating the attack.

- DOTS opportunistic-capable router: a router with the capacity to decode the opportunistic DOTS signal and re-embed such an information in other IPv6 packets.

- All DOTS opportunistic-capable agents are defined as the DOTS agents supporting the opportunistic DOTS signal processing.

2. Asynchronous DOTS signaling

2.1. Motivation

The traffic generated by a DDoS can be characterized according to various parameters, such as the layer (IP/ICMP or application), maximum and instant throughput, among others. Regardless its nature, we assume that for most cases, a DOTS client will be able to signal back one or few messages, during the attack, to the DOTS phase.

We have the same behavior in other DDoS attacks. For instance, on November 30th and December 1st, 2015, the Root DNS system was hit by an application layer (DNS) attack [rootops-ddos]. Each one of the 13 root server letters (A-M) was hit by attacks peaking at 5 million queries per second. By utilizing the RIPE Atlas DNSMON infrastructure, we can see that during the DDoS attacks, most of the root server letters remained reachable and able to respond to the DNS request sent by the probes employed by the DNSMON [ripe-dnsmon-ddos]. Few letters, however, had a packet loss rate of more than 99%. The DNSMON probes, however, experience mostly delays in their DNS requests instead.

As regular signaling from the DOTS client to the DOTS server or the DOTS gateway might be affected by the attack traffic, it is important to maximize the delivering success of the signals by using alternative packets and/or paths to deliver it. As a result, it forces to have intermediate agents, DSRs, able to catch DOTS signals delivered through those auxiliary mechanisms. However, those agents MUST not always forward DOTS signal to the server in order to limit the induced overhead. Only if the regular signal is not received by

the server, retrieving the signal from the DSRs is required, which has thus initiated by the DOTS server itself.

2.2. Architecture

DSR (DOTS Signal Repository) are additional REQUIRED datastores. They are integrated in the DOTS architecture [I-D.ietf-dots-architecture]as highlighted in Figure 1. (1) refers to the regular signaling. (2) and (3) refers to the proposed auxiliary mechanism. As shown in this figure, DSRs act as synchronizing agent where the DOTS client drops signal information (2) (attack details). On the contrary, the server will retrieve this information (3).

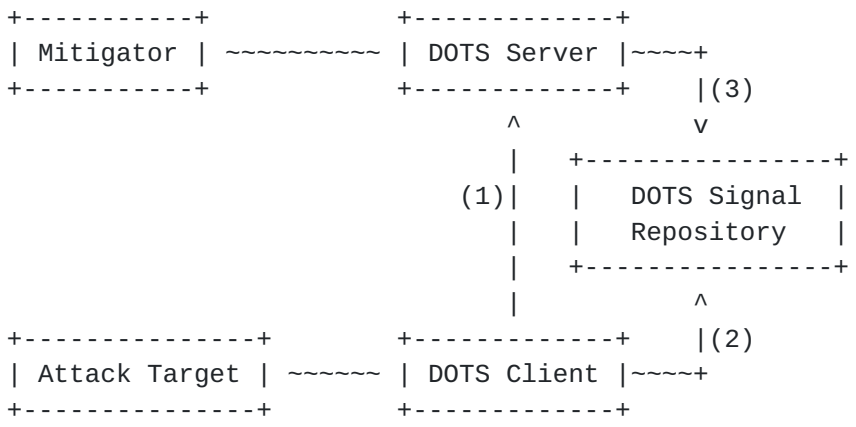


Figure 1: Asynchronous signaling

2.3. Asynchronous process

This section describes the process of asynchronous DOTS signal processing. In Figure Figure 1, there are two main communication channels which are not synchronized:

- Between DOTS client and DSR: the client sends DOTS signal information when a condition is satisfied. The condition MUST be configured by the user but SHOULD be linked to information provided by Attack target. For instance, when an attack is detected, the client connects to DOTS server and in parallel sends signal to one or more DSRs.

- Between DOTS server and DSR: the server will retrieve signal information when a specific condition occurs. Such a condition is linked with the probable occurrence of an attack. The server can infer this condition when the client is not responsive anymore. In [I-D.ietf-dots-signal-channel], an heartbeat mechanism is defined. Hence, when no heartbeat is received from the client, the server MUST try to get signal information by the asynchronous communication channel.

Each communication channel can implement its own protocol. They are NOT REQUIRED to be the same.

We have to note that the client condition to provide signals to the DSR can be weaker than regular synchronous signaling between DOTS client and server. Indeed, a client can signal to the DSR some suspect activities for which no mitigation is required yet. However, when the supposed attack is stronger provoking client disruption, the latter is not able to provide any type of signaling anymore and the server can thus retrieve information on prior stored signals.

2.4. Protocol requirements

DOTS signaling requirements are documented in [I-D.ietf-dots-requirements].

GEN-003 (Bidirectionality) requires that signal channel MUST enable asynchronous communications between DOTS agent by allowing unsolicited messages. Asynchronous signaling described in the current document allows DOTS client to provide signals, which can be retrieved later by DOTS server(s).

Because of this mechanism, there are requirements which are not supported: OP-002 (Session Health Monitoring), OP-003 (Session Redirection). Therefore, the fall-back asynchronous DOTS signaling is an additionnal mechanism and MUST NOT replace regular signaling as described in [I-D.ietf-dots-signal-channel].

It is particularly designed to fulfill GEN-002 (Resilience and Robustness) by increasing the signal delivery success even under the severely constrained network conditions imposed by particular attack traffic.

In addition, the fall-back asynchronous DOTS signal MUST specify a TTL (Time-to-Live) used by DSRs to store received signal in a limited period of time. It is different from mitigation lifetime but MUST be lower or equals.

3. DOTS Signal Repository

DSRs have to be deployed and distributed in order to enhance its reachability by the DOTS client. It is NOT REQUIRED that all DOTS agents use the same set of DSRs and a DOTS client and server SHOULD define their own set regarding their particular context, e.g. the network topology. The Data channel [I-D.ietf-dots-data-channel] between client and server MUST be used to configure it. As an example in the case of the inter-domain scenario, the DOTS server can inform the DOTS client to use DSRs scattered in multiple domains.

There is no restriction on the environment where DSRs can be deployed. Two types of DSRs are mainly considered:

- Routers: they have low capacity to process and store received signals but they are well distributed by nature in the network.

- Servers or stations: they provide higher computational power and storage but are less distributed

Selection of protocols to use for asynchronous signaling MUST take into account those specificities.

4. Protocol

4.1. Between DSR and DOTS server

To retrieve signals, the DOTS server MUST request the DSRs. Standard protocols can be used. Requests from the DOTS server MUST specify a client identifier and the DSRs returns all stored signal related to this client. The protocol MUST provide integrity and authenticity and SHOULD guarantee confidentiality. To limit entailed overhead lightweight protocol SHOULD be used. COAP [RFC7252] over DTLS [RFC6347] is RECOMMENDED when DSRs are servers. In the case of routers acting as DSR, network management protocol such as SNMP [RFC1157] or NETCONF [RFC6241] SHOULD be leveraged.

4.2. Between DSR and DOTS client

The signal sent by the DOTS client to the DSR is more prone to be affected by attack traffic due to its proximity to the attack victim. Similar protocol as between DSR and DOTS server can be used but it is RECOMMENDED to convey the signal over multiple paths to increase the reachability sucess

This document introduces two mechanisms to deliver the signal from the DOTS client to the DSR: IPv6 opportunistic signaling using Hop-by-Hop Option header; source routing using IPv6 Segment Routing Header (SRH).

4.2.1. Opportunistic DOTS signaling

This section specifies a signalling mechanism that instead of designing a new application-layer protocol, it utilizes the IPv6 Hop-by-Hop header [RFC2460]. This header SHOULD be processed by intermediate devices and it MUST be the first header in IPv6 extension headers [RFC7045].

In such a particular scenario, DSRs are intermediate routers capable of processing the option. DOTS server MAY also receive IPV6 packets with the Hop-by-Hop option and could thus process it directly. It is till considered as asynchronous since the server MAY NOT receive the initial packet emitted by the client but a copy of the signal in another packet (done by an intermediate DSR/router). Otherwise, it MUST connect to the DSR (section Section 4.1)

The new option containing the attributes of the signalling message is included in an opportunistic way in available IPv6 packets leaving a network element. The DOTS client will thus embed the signalling attributes into outgoing IPv6 packets not necessarily going to the DOTS server. Intermediate routers receiving such a packet will examine it and embed the same information into other IPv6 packets. domain in this opportunistic way to increase the probability that such a packet will be finally forwarded to a DOTS gateway or Server, but also in controlled way to avoid that the mechanism is exploited for a malicious purposes.

Only the Hop-by-Hop options header allows such behavior and using Destination options header is not enough to make the DOTS signal going through the network in an opportunistic way. Each network element recognizing this new option will select the best fitted IPv6 packets to deliver the signal to the DOTS DSRs. For this reason the Hop-by-Hop header option is essential to make such behavior compared to other existing IPv6 extension headers [RFC6564].

The goal is to provide an efficient mechanism where nodes in a IPv6 network facing a DDoS attack can deliver a DOTS signal message sent by a DOTS client to the DOTS server. The specified mechanism does not generate transport packets to carry the DOST signal message but it only relies on existing IPv6 packets in the network to include inside them a hop-by-hop extension header which contains an encoded DOTS signal message. The solution defines a new IPv6 Hop-by-Hop header option with the semantic that the network node SHOULD include

the option content within one or multiple outgoing IPv6 packets available in that network node.

4.2.1.1. Hop-by-Hop option encoding

According to [RFC2460], options encoded into the IPv6 Hop-by-Hop header are formatted as Type-Length-Values (TLVs). The option for opportunistic DOTS signal is thus defined as described in Figure 2

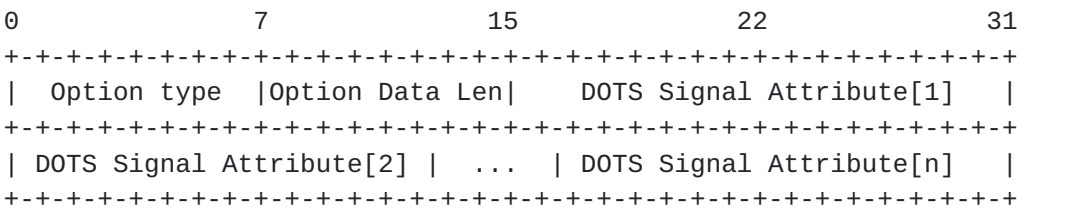


Figure 2: Hop-by-Hop option encoding

The first byte defines the Hop-by-Hop Option type number allocated to the DOTS opportunistic signalling. This number is not yet fixed but the first three bits MUST be set to 0. The first two zero bits indicate that routers which cannot handle the DOTS signal option will continue to process other options. The third 0 bit means that the option processing will not change the packet's final destination [RFC2460].

The second byte contains the length of the option content. The content of the DOTS Signal option is a variable-length field that contains one or more type-length-values (TLV) encoded DOTS signal attributes, and has the format described in Figure 3.

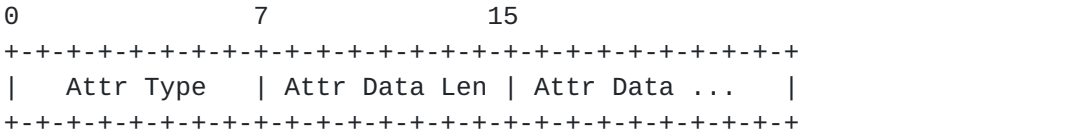


Figure 3: Hop-by-Hop option encoding

The Attr Type is 8-bit identifier of a DOTS signal attribute.

The Attr Data Len is 8-bit unsigned integer which is the length of Attr Data in bytes.

The Attr Data is variable-length field that contains the data of the attribute.

Since using TLVs in Hop-by-Hop options is known to be a factor of attacks [I-D.krishnan-ipv6-hopbyhop], DOTS attributes are encoded with fixed length when possible.

4.2.1.2. DOTS signal Option attributes

The first attribute embedded into the opportunistic DOTS signal is a TTL (Time-to-Live) field which indicates the maximum number of retransmission of the signal into another IPv6 packets until it MUST be discarded. Remaining attributes are similar to the header fields described in [I-D.ietf-dots-signal-channel] used to convey a DOTS signal through a HTTP POST.

The sequence of attributes to be inserted within the header MUST start with fixed-length attributes which are defined in the following order:

- TTL: Time-to-Live. This is a mandatory attribute encoded in one byte.

- Flags: one byte is reserved for flags.
  The first bit indicates the type of the IP address of the host: 0 for IPv4, 1 for IPv6. The second bit indicate if the protocol to use is TCP (1) or UDP (0). The third bit indicates if the message is signed. The remaining bit are not used yet.

- host: the IP address of the DOTS server where the signal option SHOULD be delivered. Depending on the flags, this field is encoded in 4 or 16 bytes.

- port: the listening port of the DOTS server. It is encoded in 2 bytes.

The remaining attributes MUST be TLV encoded, and they are defined in the following order:

- policy-id: defined in [I-D.ietf-dots-signal-channel].

- target-ip: defined in [I-D.ietf-dots-signal-channel].
  However, each address or prefix is encoded in its own TLV element. The distinction between IPv4 and IPv6 is done over the length of the value.

- target-port: defined in [I-D.ietf-dots-signal-channel].
  However, each target port is encoded in its own TLV element.

- target-protocol: defined in [I-D.ietf-dots-signal-channel]
  However each target protocol is encoded in its own TLV element.

o lifetime (lt): lifetime of the mitigation request defined in [I-D.ietf-dots-signal-channel]

The encoded attributes MUST be included in the option header in the order defined above.

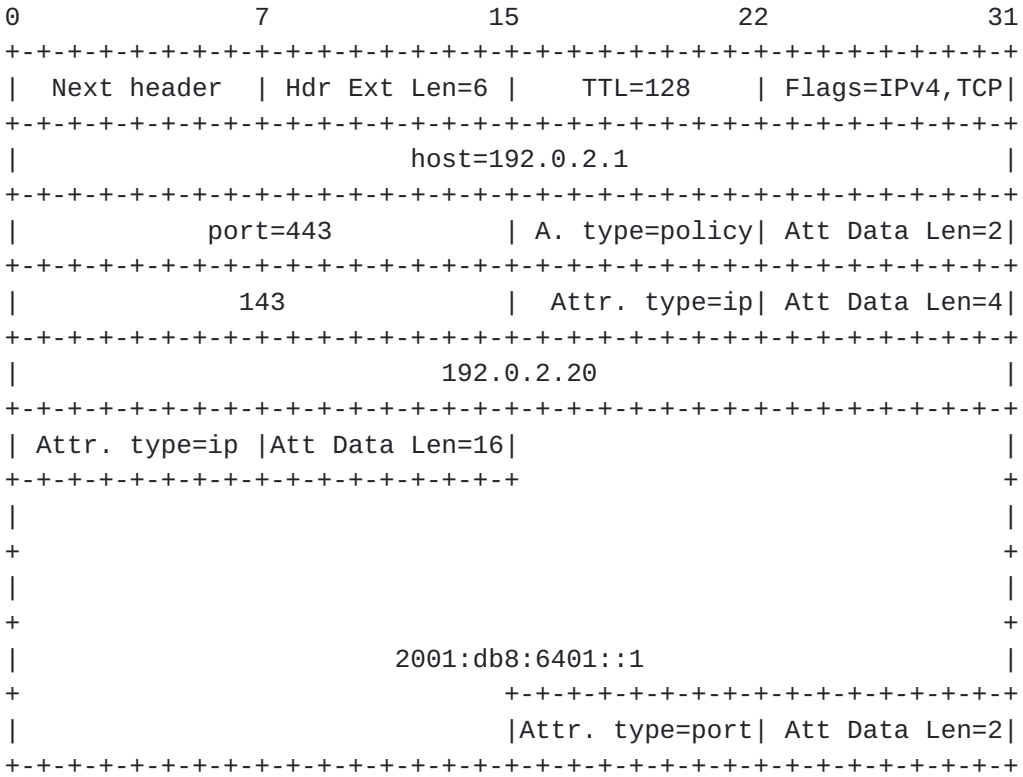
Table 1 provides the value of types that are used by the TLV encoded attributes.

Attribute type	Value
policy-id	0
target-ip	1
target-port	2
target-protocol	3
lifetime	4

Table 1: TLV encoded attributes types

4.2.1.3. Example

Following is an example of an encoded Hop-by-Hop Option header to signal that a web service is under attack.



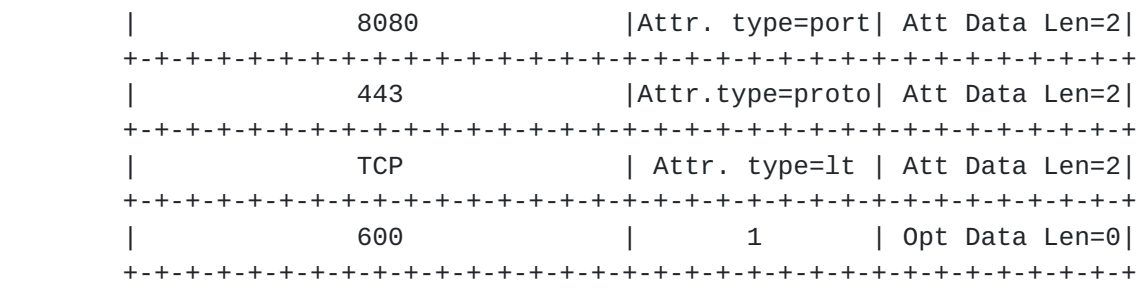


Figure 4: Example of Hop-by-Hop DOTS signal encoding

In the previous example, the message is not signed and terminates with padding. If it is the case, then the signature MUST BE added at the end such that the integrity and authenticity can be checked by the DOTS server or gateway. The TTL attributes MUST be excluded from the signature calculation.

4.2.1.4. Option Processing

4.2.1.4.1. Opportunistic DOTS signal initialization by a DOTS client

When a DOTS client needs to inform the DOTS server that it is under attack, it firstly makes a connection attempt and applies the mechanisms described in [I-D.ietf-dots-signal-channel].

In addition, it MAY activates an opportunistic mechanism to include the Hop-by-Hop header option specified in this document in one or multiple available IPv6 packets leaving the node. Because the DOTS client location is independent of the signalling, it can be positionned in a part of the network where there is no passing-by traffic which can serve for opportunistic signalling. DOTS client MAY also create and emit IPv6 datagrams without payload but with the signal encoded in the Hop-by-Hop option header.

Otherwise, the selection of packets has to be configured a priori. The configuration is composed of a sequence of rules defined in a hierarchical order such that they are triggered in a sequential manner.

The selection of packets has to be configured a priori. The configuration is composed of a sequence of rules defined in a hierarchical order such that they are triggered in a sequential manner.

Each rule is defined by:

- a set of filters over the IPv6 packet headers. Only packets matching those filters are selected for opportunistic signalling.

For instance, only packets heading to a given subnetwork or to specific address close to a DOTS server can be selected to increase the chance to reach the latter.

- a ratio to select only a proportion of packets matching the filters in order to limit the induced overhead of the opportunistic signalling.

- a timeout until the rule is active and selected IPv6 packets embed the DOTS opportunistic signal.

The objective is to apply each ordered rule after another according to their timeouts. The first rule is triggered immediately after the opportunistic signalling is activated.

In all cases (embedding information into an exsiting packet or creating an new pakket with no payload), the client MUST avoid fragmentation.

Although the definition of rules MUST be configured by the user. It is RECOMMENDED to order them inversely related to the number of packets that would be selected. This can be approximated regarding the definition of filters. The core idea is to benefit from the first instants of the attack before losing connectivity by using a maximum number of outgoing packets to include the DOTS signalling option. It is thus RECOMMENDED to define the first as matching all IPv6 packets with a ratio equals one to rapidly disseminate the information but with a short timeout to limit the implied overhead.

Here is the an example of rules:

1. all outgoing IPv6 packets with a 10 second timeout

2. all outgoing IPv6 packets with a ratio of 10% and a 1 minute timeout

3. all outgoing multicast IPv6 packets with a ratio of 10% and a 1 minute timeout

4. all outgoing IPv6 packets heading to the DOTS server with a ratio of 100% and a one hour timeout

4.2.1.4.2. Processing by a non DOTS opportunistic-capable router

When receiving an opportunistic DOTS signal encoded in a IPv6 packet, a non DOT opportunistic capable router simply skips the Hop-by-Hop option and continue the normal processing of the IPv6 packet because the option type MUST start with three zero bits.

4.2.1.4.3. Processing by a DOTS opportunistic-capable router

A DOTS opportunistic-capable router MUST store DOTS signalling information whose it is aware of. If a router processes an IPv6 DOTS opportunistic signal and supports this option, it first checks if it has already stored the associated information. In that case, the router simply skips the option and continues the normal processing otherwise it stores the encoded information in order to embed it again in other IPv6 packets similarly to the DOTS client. Hence, a set of rules are also defined in advance and are triggered upon the reception of a new opportunistic DOTS signal. Once all rule have been applied, signalling information MUST be discarded by the router. When embedding the information into other IPv6 packets, the router MUST decrease the TTL by one since opportunistic signalling does not prevent loops in the dissemination of signalling.

4.2.1.4.4. Processing by a DOTS opportunistic-capable gateway

If a DOTS gateway has DOTS capabilities, it will apply the same strategy as a DOTS client by making attempts of direct connections to the DOST server and in addition it inserts the Hop-by-Hop header DOTS signalling option in leaving IPv6 packets using the strategy specified above.

4.2.1.4.5. Processing by a DOTS opportunistic-capable server

When the IP layer of the host where the DOTS server is running receives an IPv6 packet carrying a Hop-by-Hop DOTS signal option header it MUST extracts the content of the option and provides the attributes data to the server program.

4.2.1.5. Deployment considerations

This mechanism will be potentially used by networks with IPv6 capable elements and requires that of IPv6 traffic exist in the network during the attack. The existing IPv6 traffic to be used could be of any type from management or user levels. It is also important to emphasize that while our mechanism utilizes an IPv6 header field, it can also be used to signal IPv4 attacks as well - given that the network devices are dual stacked.

IPv6 extension headers are often rate-limited or dropped entirely. To be able to use the mechanism specified in this document, network operators need to avoid discarding packets or ignoring the processing of the hop-by-hop option on their deployed network elements. However, instead of dropping or ignoring packets with hop-by-hop option carrying DOTS signal, they need to assign these packets to slow forwarding path, and be processed by the router's CPU. This

behavior will not affect the performance of the network devices since the network is already facing a DDoS attack and fast forwarding paths are saturated by the attacker traffic.

If the DOTS server, gateway and the client are located in the same administrative domain, marking the IPv6 packets with the proposed hop-by-hop header option could be done in a straight forward way, while considering that an agreement exists inside the domain to avoid dropping or rate limiting of IPv6 extension headers as described above. The proposed mechanism becomes less practical and difficult to deploy when the DOST server is running on the Internet. In such scenario, the mechanism could be used in the intra-domain part to deliver the hop-by-hop option carrying the DOTS signal until it reaches a DOTS gateway located in the same domain as the client, then the gateway will apply mechanisms provided by the DOTS transport protocol [I-D.ietf-dots-signal-channel] to inform the server running on Internet about the attack. This deployment scenario requires that at least one DOTS gateway is deployed in the same domain than the DOTS client.

4.2.1.6. Impact on existing IP layer implementations

For this option to be applicable within an IP system, it requires modifications to existing IP layer implementation. At DOTS capable nodes (client, gateway and server), it requires a service interface used by upper-layer protocols and application programs to ask the IP layer to insert and listen to the Hop-by-Hop header option in IPv6 packets. A DOTS client invokes the service interface to insert the option, A DOTS gateway invokes the service interface for listening and inserting the option, and finally a DOTS server only invokes the service interface to listen to the DOTS signalling option.

Intermediate nodes (routers or middle boxes) IP layer needs to be extended to perform processing of the new Hop-by-Hop header option. They mainly parse the first host attribute of the option and make a selection of a leaving IPv6 packet where the option will be inserted.

Every node inserting the new proposed Hop-by-Hop option SHOULD only select IPv6 packets with enough left space to avoid fragmentation.

4.2.2. IPv6 SRH

DOTS signalling may be carried using IPv6 source routing header. Details will be provided in a later version of this document.

5. Security Considerations

Any IPv6 header option could be used by an attacker to create an attack on the routers and intermediate boxes that process packets containing the option. The proposed IPv6 option in this document MAY be abused by an attacker to create a covert channel at the IP layer where data is hidden inside the content of the option [RFC6564]. However, this attack is not specific to the proposed option and it is a known issue of IPv6 header extensions and options. The option MAY also be used by an attacker to forge or modify opportunistic DOTS signal leading to trigger additional processing on intermediate nodes and DOTS servers.

However the proposed option should be only initiated by a DOTS client and information embedded in new IPv6 messages by opportunistic DOTS capable routers. Defining proper policies to filter all messages with this option set and originated from other nodes would limit security issues since these DOTS opportunistic-capable agents SHOULD be trustworhy.

In addition, the message MAY be signed using techniques to enforce authenticity and integrity over the opportunistic DOTS signal channel. The signalling message specification includes a flag to indicate if the message is signed by the choice of the signature algorithm is let to the users. This signature has to be computed by the DOTS opportunistic-capable client and checked by the DOTS opportunistic-capable gateway or router. Hence, intermediate routers MUST NOT modify the message and its signature except the TTL, which so has not be considered during the signature computation.

Assuming a compromised router, the attacker could nevertheless replay the message or increase the TTL but thanks to the unique policy-id all intermediate-DOTS capable router will drop such messages and thus limiting their forwarding in the network.

Besides, an attacker can also listen opportunistic DOTS signals to monitor the impact of its own attack. These considerations are not specific to the proposed option and supposes that the attacker is able to compromise intermediate routers.

6. IANA Considerations

This draft defines a new IPv6 hop-by-hop option[RFC2460].This requires an IANA RFC3692-style update of:http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml and ultimately the assignment of a new hop-by-hop option according to the guidelines described in [RFC5237].

7. Acknowledgements

This work is partly funded by FLAMINGO, a Network of Excellence Seventh Framework Programme.

8. References

8.1. Normative References

[I-D.ietf-dots-architecture]
Mortensen, A., Andreasen, F., Reddy, T., christopher_gray3@cable.comcast.com, c., Compton, R., and N. Teague, "Distributed-Denial-of-Service Open Threat Signaling (DOTS) Architecture", draft-ietf-dots-architecture-01 (work in progress), October 2016.

[I-D.ietf-dots-data-channel]
Reddy, T., Boucadair, M., Nishizuka, K., Xia, L., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel", draft-ietf-dots-data-channel-00 (work in progress), April 2017.

[I-D.ietf-dots-requirements]
Mortensen, A., Moskowitz, R., and T. Reddy, "Distributed Denial of Service (DDoS) Open Threat Signaling Requirements", draft-ietf-dots-requirements-04 (work in progress), March 2017.

[I-D.ietf-dots-signal-channel]
Reddy, T., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel", draft-ietf-dots-signal-channel-01 (work in progress), April 2017.

[I-D.krishnan-ipv6-hopbyhop]
Krishnan, S., "The case against Hop-by-Hop options", draft-krishnan-ipv6-hopbyhop-05 (work in progress), October 2010.

8.2. Informative References

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, <http://www.rfc-editor.org/info/rfc1157>.

[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>.

[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>.

[RFC5237] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the Protocol Field", BCP 37, RFC 5237, DOI 10.17487/RFC5237, February 2008, <http://www.rfc-editor.org/info/rfc5237>.

[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>.

[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>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <http://www.rfc-editor.org/info/rfc6564>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <http://www.rfc-editor.org/info/rfc7045>.

[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>.

[kuhrer2014exit] M. Kuhrer, T. Hupperich, C. Rossow, T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks", USENIX Security Symposium 23rd, 2014.

[ripe-dnsmon-ddos] RIPE, "NCC DNS Monitoring Service (DNSMON)", <https://atlas.ripe.net/dnsmon/>.

[rootops-ddos] rootops., "Events of 2015-11-30", 2015, <http://root-servers.org/news/events-of-20151130.txt>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Jerome Francois
Inria
615 rue du jardin botanique
Villers-les-Nancy 54600
FR

Phone: +33 3 83 59 30 66
Email: jerome.francois@inria.fr

Abdelkader Lahmadi
University of Lorraine - LORIA
615 rue du jardin botanique
Villers-les-Nancy 54600
FR

Phone: +33 3 83 59 30 00
Email: Abdelkader.Lahmadi@loria.fr

Marco Davids
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: marco.davids@sidn.nl

Giovane Moura
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: marco.davids@sidn.nl