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.

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 analyzer, 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:
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 Nishizuka, et al. Expires August 22, 2016

----------
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:

- 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.

- 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.

- 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:

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.
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:

- 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.

- 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.

- 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:
Provisioning stage: Before getting attacked by malicious traffic, a DOTS client needs to register to the DOTS server, as well as enable capacity building in advance;

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:

- 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;

- 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;

- 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 comply with the negotiated contract;
thresholds_pps: If an attack volume is over this threshold, the controller will reject the protection in order to comply with the negotiated contract;
duration: If an attack longed over this threshold, the controller will reject the protection in order to comply 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:
4.1.2. Operations

The main operations in the provisioning stage include:

- The customers (DOTS client) registers to ISP controller with capability building including protection methods, process capacity, IP address scope, deployment position, etc;
- 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;
- 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 (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:
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
  }

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:
4.2.2. Operations

The main operations in the signaling stage include:

- The customer (DOTS client) detects malicious attack, requests mitigation service to its ISP controller (DOTS server);
- ISP controller authenticates the customer and provides its intra-domain mitigation service to customer;
- 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;
- 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;

- Working ISP controllers reports its statistics result by mitigation status request message to the orchestrator for counting purpose;

- The customer can updates its mitigation scope to the ISP controller. It also can notify its mitigation efficacy result to the ISP controller;

- When the ISP controller is informed from the mitigator that the attack is over, it should notify the customer to terminate the mitigation service;

- The heartbeat message is exchange 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


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

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