DOTS Data Channel

DOTS Interim Meeting, February 2018

Presenter: Mohamed Boucadair
Agenda

• Quick walk through the protocol
• Pending Issues
• Next steps
Design Walk Through

• DOTS agents use RESTCONF for:
  – Managing aliases
  – Managing filtering entries

• The design avoids requiring modifications to RESTCONF

• Re-use the following parameters as defined in the signal-channel specification:
  – cuid: clients MUST use the same 'cuid' for both signal and data channels
  – cdid

• Assume a default direction for filtering rules
  – DOTS client domain is the “destination”

• Resources quota is configured on the server to:
  – Limit the number of alias/filtering entries to be instantiated by a DOTS client/domain
  – Clients will be granted resources only if the quota is not reached
This structure allows to achieve all the required DOTS operations on aliases:
Create an alias, retrieve aliases of a given client, delete an alias of a client, etc.
Design Walk Through

RESTCONF Constraints & DOTS Requirements

draft-ietf-dots-data-channel-12

augment /ietf-acl:access-lists/ietf-acl:acl:
    +++-rw cuid                      string
    +++-rw client-domain-hash?      string
    +++-rw lifetime                int32
augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
    /ietf-acl:ace/ietf-acl:actions:
        +++-rw rate-limit?    decimal64
augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
    /ietf-acl:ace/ietf-acl:matches/ietf-acl:ipv4-acl:
        +++-rw fragments?    empty
augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
    /ietf-acl:ace/ietf-acl:matches/ietf-acl:ipv6-acl:
        +++-rw fragments?    empty
augment /ietf-acl:access-lists:
    +++-rw dots-acl-order

This one is more problematic, e.g., it does not allow:
- To retrieve an ACL of a given client
- To retrieve all ACLs of a given client
- To delete an ACL of a given client
- To delete all ACLs of a given client
A new structure is adopted to honor DOTS requirements, while leveraging on existing specifications:

- Mimic the same structure of ACLs as defined in netmod.
- Rely upon matching criteria defined by netmod WG; a DOTS profile is defined,
- Define new actions (e.g. rate-limit)
- Support fragments
Design Walk Through

Register a DOTS Client

```
POST /restconf/data/ietf-dots-data-channel:dots-data HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
    "ietf-dots-data-channel:dots-client": [
        {
            "cuid": "dz6pHjaADkaFTbjr0JGBpw"
        }
    ]
}
```

```
PUT /restconf/data/ietf-dots-data-channel:dots-data\/dots-client=dfrtAA78yFGhouxgioulmxw HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
    "ietf-dots-data-channel:dots-client": [
        {
            "cuid": "dfrtAA78yFGhouxgioulmxw"
        }
    ]
}
```
Design Walk Through

Server-domain Gateway addition of 'cdid' for DOTS client registration

POST /restconf/data/ietf-dots-data-channel:dots-data HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dz6pHjaADkaFTbjr0JGBpw",
      "cdid": "7eeaf349529eb55ed50113"
    }
  ]
}

PUT /restconf/data/ietf-dots-data-channel:dots-data/\n  /dots-client=dfrtAA78yFGrhuxgioulmxxw HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dfrtAA78yFGrhuxgioulmxxw",
      "cdid": "7eeaf349529eb55ed50113"
    }
  ]
}
Design Walk Through

DOTS Operations

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>cuid</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an alias/filter</td>
<td>POST or PUT</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Update an alias/filter</td>
<td>PUT</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Delete an alias/filter</td>
<td>DELETE</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Retrieve installed aliases/filters</td>
<td>GET</td>
<td>Mandatory</td>
<td>Optional</td>
</tr>
</tbody>
</table>

- ‘cdid’ is not required to be injected by server-side gateways once a client is registered.
- It is the responsibility of the server to maintain ‘cdid’/’cuid’ associations.
- DOTS servers returns “403 Forbidden” if ‘cdid’ is not present.
Design Walk Through

De-register a DOTS Client

DELETE /restconf/data/ietf-dots-data-channel:dots-data/
dots-client=dz6pHjaADkaFTbjr0JGBpw HTTP/1.1
Host: {host}:{port}

All resources bound to this client will be removed by the server
Issue #1: Avoid Stale Mappings

DELETE /restconf/data/ietf-dots-data-channel:dots-data/dots-client=dz6pHjaADkaFTbjr0JGBpw HTTP/1.1
Host: {host}:{port}

All resources bound to this client will removed by the server

Some “misbehaving” clients may not clean correctly their entries

Stale entries are therefore likely
Issue #1: Avoid Stale Mappings (Option 1)

A lifetime hint is included in the resource creation request.

The server may honor the suggested lifetime or assigns a distinct value as per its local policies.

How to notify the client about the assigned value?

RFC8040: "If the POST method succeeds, a "201 Created" status-line is returned and there is no response message-body."
Issue #1: Avoid Stale Mappings (Option 2)

Servers MUST maintain an entry for a minimum period (e.g., 1 week, 1 month). No Lifetime is included in a request.

If no refresh from the client, the server removes expired entries. This prevents against amnesic clients.

Suggested position: Recommended
Issue #2: Filter Activation

• Do we assume that all filtering rules are activated by default or only when a mitigation is active?

• Proposal:
  – We should *support both*
  – The intended action is governed by a new attribute called “activation-type” which can be set to “immediate” or “mitigation-time”
  – Which default value to use?

• Comments?
Issue #3: Filter Direction

• Do we need to support explicit “direction” in filtering rules: “in”/“out”?  

• Proposal:  
  – The current default direction is aligned with the nature of DDoS attacks targeted by DOTS (incoming)  
  – Direction can be defined as an extension (if needed)  
  – *No text change is required*  

• Comments?
Issue #4: Global or Per-client Filters?

• Do we consider filters created by a client are globally available, or just for the client?

• Proposal:
  – *Filters are global*
  – It is the responsibility of the client domain to ensure consistency
  – Text to be added to make it clear

• Comments?
Issue #5: Filtering Fields

• Should we supporting all of the fields as defined by “ietf-packet-fields”?
  – Do we need to define a minimum supported set?

• Proposal:
  – Adhere to what is indicated in ietf-packet-fields
    – *No text change is needed*

• Comments?
Issue #6: Address Change

• DOTS servers MUST verify that a DOTS client is entitled to enforce filtering rules on a given IP prefix
  – What happens if that prefix is not attached to the DOTS client domain anymore (e.g., renumbering)?

• Proposal:
  – Servers MUST NOT maintain a filtering entry beyond the lifetime, unless the client asked to refresh it
  – Clients MUST update their filtering entries upon change of the destination-prefix

• Comments?
Next Steps

• Publish -14 to fix Issues #1, #2, #4, and #6
• Any issues that are not covered?
• Questions?
Appendix
Design Walk Through

Avoid Stale Mappings (option 3)

No lifetime is associated with alias/filtering entries, but a check is made at the DOTS client level: If a client is not inactive since X weeks, consider its entries as stale ones.

Is it safe to delete all filtering rules?
Design Walk Through

Avoid Stale Mappings (option 3, revised)

No lifetime is associated with alias/filtering entries, but a check is made at the DOTS client level: If a client is not inactive since X weeks, consider its entries as stale ones.

Inactive filters can be safely removed. Activated filters are migrated to other DOTS clients of the same domain. Those will need to clean, if needed.