DOTS
Interop Report

IETF 102 Hackathon
Kaname Nishizuka/NTT Communications
Jon Shallow/NCC Group
Liang Xia/Huawei
# Hackathon Plan

<table>
<thead>
<tr>
<th>Hackathon</th>
<th>What we did</th>
<th>Signal Channel</th>
<th>Data Channel</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF99</td>
<td>Implementation of OSS (go-dots)</td>
<td>✔️</td>
<td></td>
<td>NTT</td>
</tr>
<tr>
<td>IETF100</td>
<td>1st Interoperability Test</td>
<td>✔️</td>
<td></td>
<td>NTT, NCC Group, Huawei</td>
</tr>
<tr>
<td>IETF101</td>
<td>2nd Interoperability Test</td>
<td>✔️</td>
<td></td>
<td>NTT, NCC Group, Huawei</td>
</tr>
<tr>
<td>IETF102</td>
<td>3rd Interoperability Test(*)</td>
<td>✔️</td>
<td>✔️</td>
<td>NTT, NCC Group, Huawei</td>
</tr>
<tr>
<td></td>
<td>- The first data-channel interop</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) NTT & NCC Group did two internal interop tests before the IETF102 hackathon in preparation.
Interop test settings

**Signal Channel**
Session Configuration/Mitigation Request

**Data Channel**
Register Client/Aliases/ACLs
Capability Discovery

**DDoS Mitigation Action**

**go-dots**
DOTS client/DOTS server

**NCC Group**
DOTS server/DOTS client

**Signal Channel Layers**
- DOTS
- CoAP
- TLS
- DTLS
- TCP
- UDP
- IP

**Data Channel Layers**
- DOTS
- RESTCONF
- TLS
- TCP
- IP

**DDoS**
Mitigation
Action
# Interop Results

## Interoperability Testing Results

<table>
<thead>
<tr>
<th>#</th>
<th>DOTS Signal Channel Features implementation status</th>
<th>client: nttdots</th>
<th>client: ncc</th>
<th>server: ncc</th>
<th>server: nttdots</th>
<th>issue to WG</th>
<th>memo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Session Configuration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>*1 unspecified value treatment-&gt;fixed</td>
</tr>
<tr>
<td>2</td>
<td>Mitigation Request</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>▲ *2</td>
<td>#1</td>
<td>*2 has bug on invalid prefix procession (nttdots)-&gt;fixed</td>
</tr>
<tr>
<td>3</td>
<td>CoAP Ping</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>4-1</td>
<td>Observe (Mitigation)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>4-2</td>
<td>Observe (Session Config)</td>
<td>✓</td>
<td>✓</td>
<td>▲ *3</td>
<td>▲ *5</td>
<td></td>
<td>*3 no notifications-&gt;fixed *5 - get back multiple status = 6 when times out</td>
</tr>
<tr>
<td>5</td>
<td>Efficacy Update</td>
<td>✓</td>
<td>✓</td>
<td>▲ *4</td>
<td>✓</td>
<td>✓</td>
<td>*4 no lifetime refresh-&gt;fixed</td>
</tr>
<tr>
<td>6</td>
<td>Request Conflict Handling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Conflict Notify</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Deadman's Trigger</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Gateway Function</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Redirection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Happy Eyeballs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

* supporting both PKI and PSK

Huawei's implementation is based on nttdots

## DOTS Data Channel Features implementation status

<table>
<thead>
<tr>
<th>#</th>
<th>DOTS Data Channel Features implementation status</th>
<th>client: nttdots</th>
<th>client: ncc</th>
<th>server: ncc</th>
<th>server: nttdots</th>
<th>issue to WG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Register DOTS clients</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Register Alias</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>#1,#3,#4</td>
</tr>
<tr>
<td>3</td>
<td>Register Filtering Rules</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>#2,#4</td>
</tr>
<tr>
<td>4</td>
<td>Capabilities</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Gateway Function</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Data-channel - Evaluated Features

- Register DOTS clients
- Register Alias
- Register Filtering Rules
- Capabilities

### Data-channel Features implementation status

<table>
<thead>
<tr>
<th>#</th>
<th>feature</th>
<th>ncc</th>
<th>nttdots</th>
<th>arbor</th>
<th>client: nttdots</th>
<th>server: ncc</th>
<th>client: ncc</th>
<th>server: nttdots</th>
<th>issue to WG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Register DOTS clients</td>
<td>☑</td>
<td>☑</td>
<td></td>
<td>☑</td>
<td></td>
<td></td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Register Alias</td>
<td>☑</td>
<td></td>
<td>☑</td>
<td>☑</td>
<td></td>
<td></td>
<td>☑</td>
<td>#1,#3,#4</td>
</tr>
<tr>
<td>3</td>
<td>Register Filtering Rules</td>
<td>☑</td>
<td>☑</td>
<td></td>
<td>☑</td>
<td></td>
<td></td>
<td>☑</td>
<td>#2,#4</td>
</tr>
<tr>
<td>4</td>
<td>Capabilities</td>
<td>☑</td>
<td></td>
<td>☑</td>
<td>☑</td>
<td></td>
<td></td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Gateway Function</td>
<td>☑</td>
<td></td>
<td>☑</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Method:
- POST/PUT/GET/DELETE
- GET
Only minor issues are found (except for fragment bit in ACL)

• Message exchange over RESTCONF/TLS is stable
• We fixed several bugs about message format on each side
• Among them, 4 issues with the draft data channel were raised during the hackathon
6.1. Create Aliases

In POST requests, at least one of the 'target-prefix', 'target-fqdn', or 'target-uri' attributes MUST be present. DOTS agents can safely ignore Vendor-Specific parameters they don't understand.

should read (as it is true for both cases)

In POST or PUT requests, at least one of the 'target-prefix', 'target-fqdn', or 'target-uri' attributes MUST be present. DOTS agents can safely ignore Vendor-Specific parameters they don't understand.
[Fixed] #3: /aliases?content=config includes pending-lifetime in the response

- “pending-lifetime” is defined as “config false”
- so, it should not be included in the response if GET to retrieve an alias is with content=config

- The text was updated to use "content=all"
[Fixed] #4: What gets returned for ?content=nonconfig

- RFC8040 – RESTCONF B.3.1 “content” Parameter
  - To retrieve only the non-configuration child resources, the "content" parameter is set to "nonconfig". **Note that configuration ancestors (if any) and list key leafs (if any) are also returned.**

- "name" of ACL/alias should be also returned because it is key leaf

- [implementation guideline] see RFC8040
#2: fragment bits representation

[Text below is the now corrected version in draft]

- **ipv4**
    The "bits" Type
    
    ```json
    "ipv4": {
        "flags": "more"
    }
    ```

- **ipv6**
    The "empty" Type
    
    ```json
    "ipv6": {
        "fragment": [null]
    }
    ```
#2: fragment bits representation

- The current draft says:

```
{
  "name": "drop-last-fragment",
  "matches": {
    "ipv4": {
      "flags": ""
    }
  },
  "actions": {
    "forwarding": "drop"
  }
}
```

- Can it be used for representation that only ‘more’ bit is 0?
  - undefined bits are checked that they are 0 (reserved and (don’t)fragment) (RFC7951 6.5. The "bits" Type‘)

- How best to represent a fragment filtering rule with netmod-acl or enhancement.
  - BGP flowspec(RFC5575) like representation for fragments?
Signal-channel - Evaluated Features

- Session Configuration
- Mitigation Request
- CoAP Ping
- Observe (Mitigation)
- Observe (Session Config)
- Efficacy Update
- Request Confliction Handling
- Confliction Notify
- Deadman's Trigger
- Gateway Function
- Redirection
- Happy Eyeballs

Core Specifications

Additional Specifications
Bugs with new features no significant issue on the draft

- Several bugs are found on both sides but fixed during or after the hackathon
- 2 issues are raised but already fixed

<table>
<thead>
<tr>
<th>#</th>
<th>DOTS Signal Channel Features implementation status</th>
<th>Interoperability Testing Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DOTS Signal Channel Features implementation status</td>
<td>client: nttdots</td>
</tr>
<tr>
<td>1</td>
<td>Session Configuration</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Mitigation Request</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>CoAP Ping</td>
<td>✓</td>
</tr>
<tr>
<td>4-1</td>
<td>Observe (Mitigation)</td>
<td>✓</td>
</tr>
<tr>
<td>4-2</td>
<td>Observe (Session Config)</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Efficacy Update</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Request Conflict Handling</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Conflict Notify</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Deadman’s Trigger</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Gateway Function</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>Redirection</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Happy Eyeballs</td>
<td>✓</td>
</tr>
</tbody>
</table>

🔺 represents the buggy status at the hackathon. but it is now fixed
[Fixed] #1: Missing mandatory lifetime in example (Fig 7,8)

• lifetime is mandatory attribute in mitigation request[PUT]

• fixed editorial nits
[Fixed] #2: Signal Channel 4.5.3
Configuration Freshness and Notifications

• The DOTS server needs to be able to update Signal Channel configuration parameters
• DOTS client was doing a PUT to refresh configuration – overwriting any DOTS server update
• Solution was for DOTS client to do a GET to refresh configuration within the Max-Age time – Draft signal channel now updated
Takeaway

• Successful interop tests are continuing
• Core specification of DOTS is now mature enough
  – both signal-channel and data-channel
  – several issues are found (not so significant) during the hackathon
  – they’ve been sorted out and fixed
• Implementations (more implementations are welcome)
  – One OSS
    • https://github.com/nttdots/go-dots
  – One proprietary
    • NCC Group
Questions
Or
Comments?
Thank You