DOTS
First Interoperability Test

IETF 100 Hackathon Report
Kaname Nishizuka/NTT Communications
Jon Shallow/NCC Group
Liang Xia/Huawei
DOTS is now working!

- DOTS WG is aiming to make it standardized in this year
- Now we have several individual implementations
  - go-dots (open-sourced project) from NTT
  - NCC Group’s proprietary implementation
- This first interoperability test at the hackathon is a giant step for proving it works.
What happened in the Hackathon

• 3 active projects with 7 participants
  – include 3 remotely from Tokyo, London, Nanjing
• 3 Projects are:
  1. First Interoperability test of 2 individual implementations
  2. Adding new features and extensions to the open-sourced implementation
  3. (Integration with a detection system of Mirai botnet)
We won an award!

- Best Open Source Project
1. First Interoperability test of 2 individual implementations
   - go-dots (open-sourced project) from NTT
     • Kaname Nishizuka, Takahiko Nagata (Remote)
   - NCC Group’s proprietary implementation
     • Jon Shallow (Remote)
## Result of the Interop Test

**Purpose:** Check interoperability of the messages on the signal channel

<table>
<thead>
<tr>
<th>Item #</th>
<th>Messages</th>
<th>CoAP Method</th>
<th>Interop Testing (client -&gt; server)</th>
<th>Internal Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>go-dots -&gt; ncc</td>
<td>ncc -&gt; go-dots</td>
</tr>
<tr>
<td>1</td>
<td>Mitigation Request</td>
<td>PUT</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>Mitigation Request Withdraw</td>
<td>DELETE</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>Mitigation Request Status</td>
<td>GET</td>
<td>✔</td>
<td>▲</td>
</tr>
<tr>
<td>4</td>
<td>Mitigation Request Status All</td>
<td>GET</td>
<td>✔</td>
<td>▲</td>
</tr>
<tr>
<td>5</td>
<td>Mitigation Status Notify</td>
<td>observe</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>Efficacy Update</td>
<td>PUT</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>Session Configuration</td>
<td>PUT</td>
<td>✔</td>
<td>▲</td>
</tr>
<tr>
<td>8</td>
<td>Session Configuration Delete</td>
<td>DELETE</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>9</td>
<td>Session Configuration Retrieve</td>
<td>GET</td>
<td>✔</td>
<td>▲</td>
</tr>
<tr>
<td>10</td>
<td>Heartbeat</td>
<td>COAP ping</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
What we proved in the Interop

- We can start and handle a mitigation from each client over DOTS signal-channel (CoAP over DTLS)
- Plus, NCC Group’s implementation can act as a DOTS relay (gateway), so we proved that relayed mitigation requests can work over multiple organizations.
General Feedback to DOTS WG

• Implementation Experiences
  – For example most of the code modification was related to encode/decode of CoAP mapping
  – there were many implicit specifications we need to figure out and agree on
• Need more description of the content and code
• approx. 60% of the signal-channel spec has been proved to work
  – The rest will be done at/by the next IETF
go-dots Feedback to DOTS WG

• Preparation for the interop test
  – Agree on port number(-06) and URI path(-07)
  – Fixed CBOR mapping
  – Updated data models

• Code Updates during Hackathon
  – Omit empty(NULL) entries in requests
  – Fixed response body

• Test scenarios should be listed and shared
  – to get every patterns of request/response type and see normal/error behavior
  – unintended behavior can be found only by interop
NCC Group Feedback to DOTS WG (Pt 1)

• Code Updates during Hackathon
  – CBOR <-> JSON mapping fixes for NULL entries
  – Remove NULL entries confusion and deleted NULL entries in any response
  – Added support for multiple mitigation requests within a single PUT

• NCC DOTS Client crashing go-dots DOTS server
  – Disabled Signal Configuration requests
  – Disabled Heartbeats
  – Still go-dots server issues handling NCC client requests
    - to be worked on
NCC Group Feedback to DOTS WG (Pt 2)

• Outstanding NCC Group to be fixed
  – DOTS Client handling bad CoAP Ping responses
  – Support of GET empty requests that are not CBOR encoded

• Questions
  – Should NULL entries be allowed ?
  – Should a NULL entry of type Object be allowed when definition is Array ?
  – What should happen when lifetime = 0 is requested ?
  – Should there be support for multiple mitigation requests within a single PUT ?
Questions
Or
Comments?
2. Adding new features and extensions to the open-sourced implementation
Using DOTS Vendor-Specific Attributes for Global IP Reputation Sharing

DOTS Client

mitigation-scopes

Vendor-Specific:
attack-event * [target-ip]
  { target-ip
top-attack *[botnet-ip]
  { botnet-ip
  attack-type
  peak-traffic {bps pps}
  start-time
  period
  }
  }

DOTS Server

Mitigation Request
response

CoAP PUT
IP Reputation

CoAP Response

Global IP Reputation Database

botnet-ip1
attack-type
peak-traffic
start-time
period

botnet-ip2
attack-type
peak-traffic
start-time
period

botnet-ipN
attack-type
peak-traffic
start-time
period

mysql> select * from ip_reputation;

<table>
<thead>
<tr>
<th>id</th>
<th>customer_id</th>
<th>botnet_ip</th>
<th>attack_type</th>
<th>peak_traffic</th>
<th>start_time</th>
<th>attack_period</th>
<th>created</th>
<th>updated</th>
</tr>
</thead>
</table>

5 rows in set (0.00 sec)
Using DOTS Vendor-Specific Attributes for Outbound Attack Mitigation

**DOTS Client**

**DOTS Server**

**Attack Source at Botnet-IP**

**Signal Channel**

**Outbound Attack Repository**

- `botnet-ip1`
  - `target-ip`
  - `attack-type`
  - `peak-traffic` (bps/pps)
  - `start-time`
  - `period`

- `botnet-ipN`
  - `target-ip`
  - `attack-type`
  - `peak-traffic` (bps/pps)
  - `start-time`
  - `period`

**CoAP PUT**

```
 mitigation-scope Request
   ... Vendor-Specific:
   attack-event * [target-ip]
   { target-ip
top-attack *[botnet-ip]
   { botnet-ip
   attack-type
   peak-traffic {bps pps}
   start-time
   period
   }
   }
```

**CoAP Response**

```
 response
```

**MySQL Query**

```
mysql> select * from outbound_attack;
+----+------------+--------------+----------+-------------+-------------+--------------+-------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+
| id | customer_id| botnet_ip    | target_ip| attack_type | peak_traffic | start_time   | attack_period| created  | updated  |
|----+------------+--------------+----------+-------------+-------------+--------------+--------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+
+----+------------+--------------+----------+-------------+-------------+--------------+--------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+
5 rows in set (0.00 sec)
```
Thank You