DOTS Signal Channel

DOTS Interim Meeting, February 2018

Presenter: Mohamed Boucadair
Agenda

• Quick walk through the protocol
• List of main changes since IETF#100
• Next steps
A DOTS client assigns mitigation identifiers to unambiguously identify requests bound to it: mid. 'mid' values are monotonically increasing.

How the server distinguishes among these requests?
Design Walk Through

Handling Requests with Overlapping Scopes

The server detects this request has overlapping mitigation scope with an existing one.

The mitigation request with the highest numeric 'mid' value wins.

Active mitigations
- mid=123
- mid=124
The server detects this request has overlapping mitigation scope with an existing one. This one is rejected because it has a lower ‘mid’ value.
Client Domain

Collision Avoidance

Server Domain

Design Walk Through

The same mid may be used by these clients (seen from the same IP address). How to distinguish among these requests?

Active mitigations
- cuid=sss, mid=124
- cuid=xxx, mid=124

Clients assign a globally unique identifier:

- **cuid** (Client Unique Identifier).

  cuid is stable when communicating with a DOTS server.

The server binds a mitigation request to a cuid. Conflicts can be detected and reported.

- cuid=sss, mid=124
- cuid=xxx, mid=124

The same mid may be used by these clients (seen from the same IP address). How to distinguish among these requests?

- cuid is the output of a cryptographic hash algorithm
- The cryptographic hash algorithm is SHA-256
- Input:
  - The SPKI of the DOTS client X.509 certificate, the DOTS client raw public key, or the "PSK identity" used by the DOTS client in the TLS ClientKeyExchange message
  - The output of the cryptographic hash algorithm is truncated to 16 bytes
Design Walk Through

Policy enforcement

The server needs to enforce policies (e.g., limit the number of DOTS clients per domain, limit the number of mitigations per client).

How to do that?

**cdid** (client domain identifier) is injected by the server-domain gateway to supply the identity of the source client/domain to the server.

- **cdid**=dfg, **cuid**=Hji, **mid**=123
- **cdid**=oupy, **cuid**=sss, **mid**=xyz
- **cdid**=oupy, **cuid**=xxx, **mid**=abc

client-domain-identifier

GW1

GW2

GW is trusted

Client Domain 1

Client Domain X

Server Domain
Design Walk Through

Operations & Parameters

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request mitigation</td>
<td>PUT</td>
</tr>
<tr>
<td>Adjust the scope of a mitigation</td>
<td>PUT</td>
</tr>
<tr>
<td>Refresh a mitigation</td>
<td>PUT</td>
</tr>
<tr>
<td>Retrieve mitigations</td>
<td>GET</td>
</tr>
<tr>
<td>Report efficacy update</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete mitigations</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI Parameters</th>
<th>Method</th>
<th>cdid</th>
<th>cuid</th>
<th>mid</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT</td>
<td>MUST</td>
<td>NOT</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>GET</td>
<td>MUST</td>
<td>NOT</td>
<td>Mandatory</td>
<td>Optional</td>
</tr>
<tr>
<td>DELETE</td>
<td>MUST</td>
<td>NOT</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI Parameters</th>
<th>Method</th>
<th>cdid</th>
<th>cuid</th>
<th>mid</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT</td>
<td>Optional</td>
<td>Mandatory</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td>Optional</td>
<td>Mandatory</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>Optional</td>
<td>Mandatory</td>
<td>Mandatory</td>
<td></td>
</tr>
</tbody>
</table>
Design Walk Through

Sample Mitigation Request Message

Header: PUT (Code=0.03)
Uri-Host: "host"
Uri-Path: ".well-known"
Uri-Path: "dots"
Uri-Path: "v1"
Uri-Path: "mitigate"
Uri-Path: "cuid=dz6pHjaADkaFTbjr0JGBpw"
Uri-Path: "mid=123"
Content-Type: "application/cbor"

Request as sent by the client

Request as relayed by the server-domain gateway

Order is important to ease identifying a resource

Header: PUT (Code=0.03)
Uri-Host: "host"
Uri-Path: ".well-known"
Uri-Path: "dots"
Uri-Path: "v1"
Uri-Path: "mitigate"
Uri-Path: "cdid=7eeaf349529eb55ed50113"
Uri-Path: "cuid=dz6pHjaADkaFTbjr0JGBpw"
Uri-Path: "mid=123"
Content-Type: "application/cbor"
Design Walk Through

Session Configuration

DOTS Signal Channel Session Configuration sessions

DISJOINT DOTS Signal Channel Session Configuration sessions
Design Walk Through

Session Configuration

• A DOTS client can negotiate, configure, and retrieve the DOTS signal channel session behavior with its DOTS peers
  – Heartbeat interval
  – Missing heartbeats allowed
  – Acceptable signal loss ratio
  – Automated mitigation on loss of signal (trigger-mitigation)

• The same or distinct configuration sets may be used during times when a mitigation is active and when no mitigation is active

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover Configuration Parameters</td>
<td>GET</td>
<td>Obtain acceptable (e.g., minimum and maximum values) and current configuration parameters on the DOTS server</td>
</tr>
<tr>
<td>Convey Signal Channel Session</td>
<td>PUT</td>
<td>Convey the configuration parameters for the signal channel (e.g., heartbeat interval, maximum retransmissions).</td>
</tr>
<tr>
<td>Delete Signal Channel Configuration</td>
<td>DELETE</td>
<td>Set to default values</td>
</tr>
</tbody>
</table>
Design Walk Through

Out-of-Order Configuration Requests

Clients assign session identifiers to unambiguously identify requests bound to it: **sid**. ‘sid’ values are monolithically increasing.

Which one to take into account?

Active sessions
- sid=123
- sid=589

The configuration request with the highest numeric ‘sid’ value overrides the other request.

**Header:** PUT (Code=0.03)
**Uri-Host:** "host"
**Uri-Path:** ".well-known"
**Uri-Path:** "dots"
**Uri-Path:** "vl"
**Uri-Path:** "config"
**Uri-Path:** "sid=123"

**Content-Format:** "application/cbor"

<table>
<thead>
<tr>
<th>Method</th>
<th>'sid' URI Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>PUT</td>
<td>MUST</td>
</tr>
<tr>
<td>DELETE</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Changes Since IETF#100

• Handling conflicts
• NAT considerations
• No list of client-identifiers but two identifiers instead (cuid and cdid)
• Fix and clarify request URIs to identify resources
• Multiple mitigations in the same request are forbidden
• Restructure the YANG module
• Update CBOR mapping table
Changes Since IETF#100

- Same or distinct configuration may be used during idle and mitigation-active times
- Config-interval to force a client to contact the server (changes may happen at the server side)
- SNI support
- Validation of target-prefix
- Rate-limit DOTS requests and cuid regeneration
- Clarify that signal configuration messages must not be relayed
- And many other edits to enhance the readability of the document
Early Port Allocation

• Request for 4646 was handed to the AD
• IANA suggested to apply for permanent allocation
• In progress
Pending Changes (github)

• Integrate hop-limit
• Fix some few typos

https://github.com/boucadair/draft-ietf-dots-signal-channel
Next Steps

• Publish -18 to integrate hop-limit
• Issue a WGLC on -18
• Questions?