DOTS Implementation Experiences

Jon Shallow

Introduction

- Any practical implementation project generates a “snagging list”
- These snags (or issues) come out of things not thought about, mis-interpretation of specifications, ambiguous definitions etc. – they are almost inevitable!
- A snag found is not a criticism – it is an opportunity to get things right and solutions need to be offered on how to fix it
- We learn from the snags to perfect the project – in this case the refining of (draft) RFCs for future use

DOTS drafting Challenges

- There are a large number of RFCs
  - Some with a small reference to something of consequence for DOTS
  - The RFC drafters may or may not have knowledge of these specifics – so they may not appropriately get spelt out in the drafts if they need to be referred to
- Ambiguity in definitions
  - The intent of a phrase may not come clearly across
  - What the drafter intuitively understands, the reader may not

Consequently

- Specifications need to be implemented
- Interoperability need to be checked out between different implementations

NCC Current DOTS agents state (1)

- DOTS Server Signal Channel
  - Fully functional signal & configuration
  - PKI Mutual Authentication using common CA
  - UDP IPv4 and IPv6 DTLS
  - Waiting on TLS support in libcoap
- DOTS Server Data Channel
  - Nearing completion (“alias” complete”)
  - PKI Mutual Authentication using common CA

NCC Current DOTS agents state (2)

- DOTS Client Signal Channel
  - Work in progress
  - PKI Mutual Authentication using common CA
  - UDP IPv4 and IPv6 DTLS
  - Waiting on TLS support in libcoap
- DOTS Client Data Channel
  - Work in Progress
- DOTS Gateway
  - Work in Progress

nttdots

- Good starting point
- Currently cannot use it as a reference
  - Uses /.wellknown/ in CoAP path
  - /.wellknown/core returns broken information
  - No CBOR Mapping usage
  - RESTCONF is not used in data channel

CBOR

- Used https://github.com/PJK/libcbor
- Had to write CBOR->cJSON and cJSON->CBOR to do CBOR Mappings

CoAP (1)

- Used https://github.com/obgm/libcoap
- Designed for IoT
- Lot of missing DOTS functionality
  - No PKI support
  - No TLS support (currently being worked on)
  - No configuration support for MaxRetransmit, AckTimeout and AckRandomFactor
  - Observe refresh trigger does not provide original request

CoAP (2)

- Minimal API documentation
- Code limitations
  - Missing checks for NULL variables etc.
  - Memory leaks when freeing off a server context
  - Uses fprintf() for some debugging, not coap_log()
    - So only some logging goes to syslog if using a logging_handler
- In discussions on libcoap developers list about how best to do the DOTS required PKI functionality
  - Have made some local changes to get DOTS agents up and running

Draft DOTS Signal Channel Spec (1)

- Loosely defined for what should be in Requests, and more importantly in Responses
  - Multiple interpretations could lead to many combinations that a DOTS agent needs to support
  - How is the Diagnostic Message formatted?
  - When should Diagnostic Messages be used
  - When should Response Payloads be used

  [Next draft (-04) addresses a lot of this]

Draft DOTS Signal Channel Spec (2)

- Examples do not match YANG spec
  - Container(YANG) = object(JSON) (RFC7951- 5.2)
  - List(YANG) = array(JSON) (RFC7951 - 5.4)
  - “mitigation-scope” is a container, but used as array in Figure 9
  - “scope” is list (array) and should be used for the multiple responses
  - [Same confusion is in Data Channel Spec]

[Next signal spec -04 draft addresses this]

Draft DOTS Signal Channel Spec (3)

- Session Configuration
  - Is configuration per individual session, or per DOTS Client and DOTS Server signal channel interaction?
    [Subsequently told is defined in draft-ietf-dots-requirements-04]
  - No way of finding current configuration before doing PUT to configure session
    [Draft spec -04 updated to handle this]

Draft DOTS Signal Channel Spec (4)

- Using “alias[-name]”
  - What is returned for the GET mitigation/status data – is “alias” expanded, or is it a repeat of the PUT request with “alias”?
  - What happens when the “alias” (via data channel) is changed / deleted for an active mitigation?
- Further points / questions in Email submitted to DOTS mailing list 26th Sep 2017

  [Draft spec -04 updated to handle a lot of these]

Draft DOTS Data Channel Spec

- Draft-ietf-netmod-acl-model-13 rev 2017-06-12 change:-
  "Added feature and identity statements for different types of rule matches. Split the matching rules based on the feature statement and added a must statement within each container.“
  - The feature containers (e.g. ipv4-acl) are missing from all examples for the filter rules in data channel spec
  - "acl-type“ is of form "ipv4-acl“, not “ipv4”
  - draft-ietf-dots-data-channel-03 needs updating to reflect this change

- What happens when multiple ACLs are defined
  - ACEs (rules) within ACL is “ordered-by user” in ietf-netmod-acl-model-13
  - No such ordering definition for ACL in ietf-netmod-acl-model-13
  - Overlapping ACLs can be order dependent
  - How are ACLs to be sorted?
  - Sufficient / Required / Requisite requirements?

Any Questions?

Thank You