W3C Linked Data Notifications High-Level Security Review

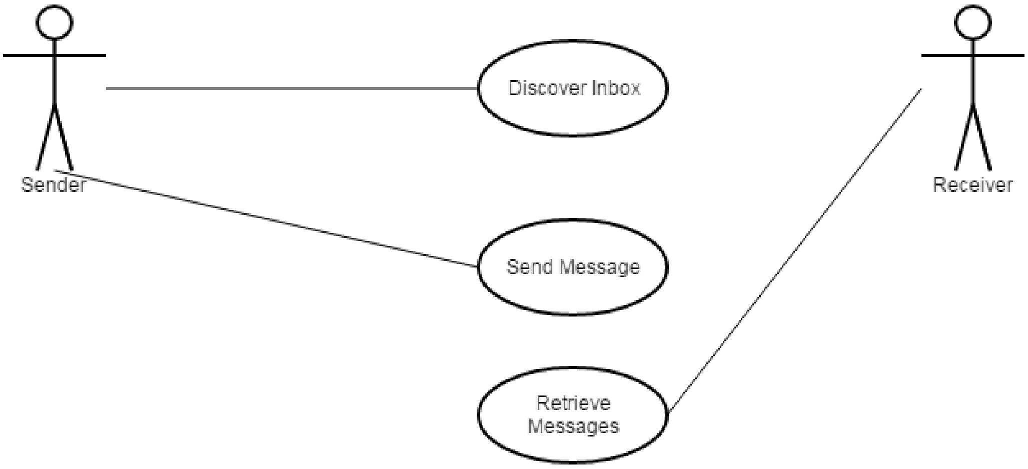
Preamble

This page contains a high-level security review of the draft W3C Linked Data Notifications (LDN) specification https://www.w3.org/TR/ldn/.

Background

The premise of the LDN specification is to enable a sender to be able to automatically discover and send a message to a recipient in an asynchronous manner.

- Discovery: Given a web resource such as a users home page on a social media website a sender can discover the owner's (recipient/s) Inbox (URL) address by making a GET or HEAD request. The result of a GET or HEAD request upon the web resource includes the address of the owner's inbox in the HTTP Link header field or via RDF.
- Sending: Given a discovered Inbox, a Sender sends a message via an HTTP POST. The Sender *may* include AuthN and or AuthZ headers to control access to the Inbox.
- Retrieving: The owner/recipients can consume messages from an Inbox via an HTTP GET request. An owner/recipient *may* include AuthN and or AuthZ headers to control access to the Inbox.



Notes

Though AuthN/Z are mentioned in respect to the sending and consuming of messages there's no mention of that during discovery. This could open a fishing attack whereby a resource linked to an inbox could be discovered without the consent of the owner. An example case could be a social network site that displays public information about the user to all. Without AuthZ'ing the consumers of this information the contact details (the linked data) for a user would be able to be harvested without the consent of the user.

There's very little information around error codes. Many "return appropriate 4xx". There's a worry that this would open up an implicit discovery path if errors are not ordered. For example, if a malicious Sender was trying to ascertain if a particular entity had an inbox at a specific service provider, the Sender could simply try and send requests to Inboxes of their choice. Now if sending a message required an AuthZ token (let's assume our malicious sender doesn't have one or it's invalid) but our receiving web-service firstly checked whether an inbox exists before the AuthX details and returned a "404 Not Found" when no inbox exists or "401 unauthorised" when it does, the Sender can ascertain valid inboxes with out successfully AuthZ'ing.

I don't see any explicit "use HTTPS" recommendation. Probably implied.

---

s3.3.3 Sender Verification Receivers should verify the sender of the notification

- by having a whitelist of senders with write access to the Inbox

- requiring authentication to enforce receiver's knowledge of every sender
- retrieving a copy of the notification from the sender's domain to verify its origin
- checking a digital signature which accompanies the notification"

I think a key worry as an end user would be spam. (A DoS issue, in security terms.) I don't think the three approaches based on sender identity really address this (global whitelist management is impractical, and spam from e.g. a cryptographically authenticated throwaway identity is still spam).

Requiring retrieval of the notification from the sender is more promising. A few observations:

- retrieving the notification immediately it is received doesn't help much - the spammer only has to maintain a working source address for a few milliseconds.
- retrieving the notification just before it is read would probably help (since the spammer has to keep its URI available for an extended period) - but at the cost of introducing user-visible delays and exposing information about the user's reading behavior to senders.

This is the same issue that large email providers face i.e. spam filtering.