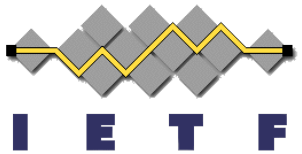
*

Poll-Based SET Token Delivery Using HTTP



draft-ietf-secevent-http-poll

Michael B. Jones

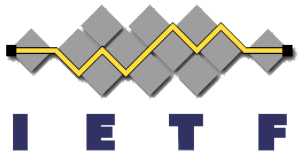
IETF 102, Montreal

July 2018

What is it?

- Defines polling delivery mechanism for SETs
- draft-ietf-secevent-http-poll is essentially draft-ietf-secevent-delivery after deletion of push-based delivery specifics
  - draft-ietf-secevent-http-push is likewise essentially draft-ietf-secevent-delivery after deletion of poll-based delivery specifics

Issues with Poll Spec (1 per slide)



Terminology Mismatch

- Terminology not aligned with SET [RFC 8417]
  - E.g., “Event Receiver” vs. “SET Recipient”

Ambiguous Normative Text

- Some normative text is ambiguous
  - E.g., description of “sets” data structure unclear

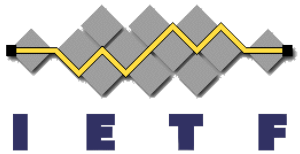
Unnecessary Duplication of Information

- The “sets” data structure contains the JWT ID (“jti”) for each SET twice
  - Once as an object name
  - Once in the object value (“jti” claim of the SET)
- Duplication introduces error possibilities that shouldn’t exist
  - What if the two “jti” values don’t actually match?
- Proposed fix:
  - Change “sets” to be simply an array of the SETs

Odd Semantics

- “maxEvents” defines “returnImmediately” to sometimes be ignored
- Parameter handling not orthogonal

Functionality without Clear Motivation



- Spec says SETs MAY be reissued
- But provides no accompanying guidance or rationale
  - Why might this occur?
  - Is it ever necessary?

Functionality Incompletely Specified

- Spec says that there SHOULD be a mechanism loss notification
  - but leaves the mechanism undefined

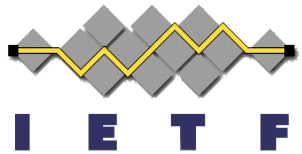
No “err” Registry

- No IANA registry is established for “err” values

Numerous Grammar and Editorial Issues

- *(not detailed here)*

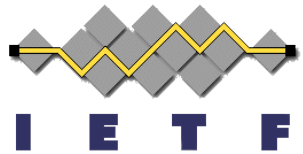
Issues Shared by Poll and Push Specs



Massive Duplication Across Poll and Push Specs

- Content of 6 of 7 top-level sections in both specs duplicated
- Push source is 708 lines, poll is 984 lines
  - 572 of these lines are identical
  - 81% identical for push, 58% identical for poll

Problems with Duplication



- Not perfectly duplicated
  - Edits to common text have already been inconsistently applied in several cases
  - Requires manual editor actions to keep in sync
  - Sometimes unclear which divergences intentional
- Some normative data structures duplicated
  - For instance, error values defined twice
  - Will they be kept in sync?
  - Will they live in a common registry?
  - Which will be authoritative?

Assertion: Current Organization Untenable

- Massive duplication creates consistency nightmare for editors
  - Manual steps to keep common text consistent
- Massive duplication creates significant work for reviewers
  - Having to figure out what's the same and what's different and try to understand why
- Massive duplication will confuse implementers
  - Having to figure out what's the same and what's different and try to understand why

Possible Solutions

- Move to three delivery specs
  - One for common pieces
  - One for only push-specific pieces
  - One for only poll-specific pieces
- Move to one delivery spec
  - All text occurs only once
  - Push-specific pieces in one section
  - Poll-specific pieces in a different section
  - Note: Both mechanisms could still be optional
  - *I believe this will be easier for all to understand*

Discussion

- As an editor, I’m not prone to fix problems in duplicated text until we solve the duplication problem
  - Unreasonable to ask us to do everything twice
- Data gathering
  - Who has reviewed both specs?
  - What was your experience reviewing them?
- How should we solve the duplication problem?
  - Should we have one or three specs?