JSON and Security

Jim Schaad
August Cellars
NOT JOSE CHAIR

Issues

- MUST have single name in any context
- Well defined JSON
- Canonicalization

Unique Names

- Current Text states
  - The names within an object SHOULD be unique.
- Problems
  - Not clear what happens if name is re-used.
- Solution
  - The names within an object MUST be unique.
  - Parser needs to error if names are not unique

Well Defined JSON

- What does a “well-defined” JSON object look like
- Leading and Trailing Whitespace
- JSON Parsers which allow garbage at the end
  - {“key”:”Value”}AAAA
  - {“key”:”value”}}
- Parsing Error vs. Parsing Ignore Problems
- Numerical precision

Canonicalization

- XML Signature has given Canonicalization a bad reputation which is undeserved
  - Characters may/may not be meaningful
    - <x>□text</x>
  - Xpath/Xquery depends on total structure
  - Schema dependencies
  - Multiple encoding formats

Canonicalization (2)

- Canonicalize and send is good
- Canonicalize, send and check is ok
- Mutual Canonicalization is problematic

Why Canonicalize

- Reduce cryptographic attack points
- Ease duplicate checking on parse
- Size reduction (no excess whitespace)
- Single string representations

Questions