

Issues in Identifier Comparison for Security Purposes

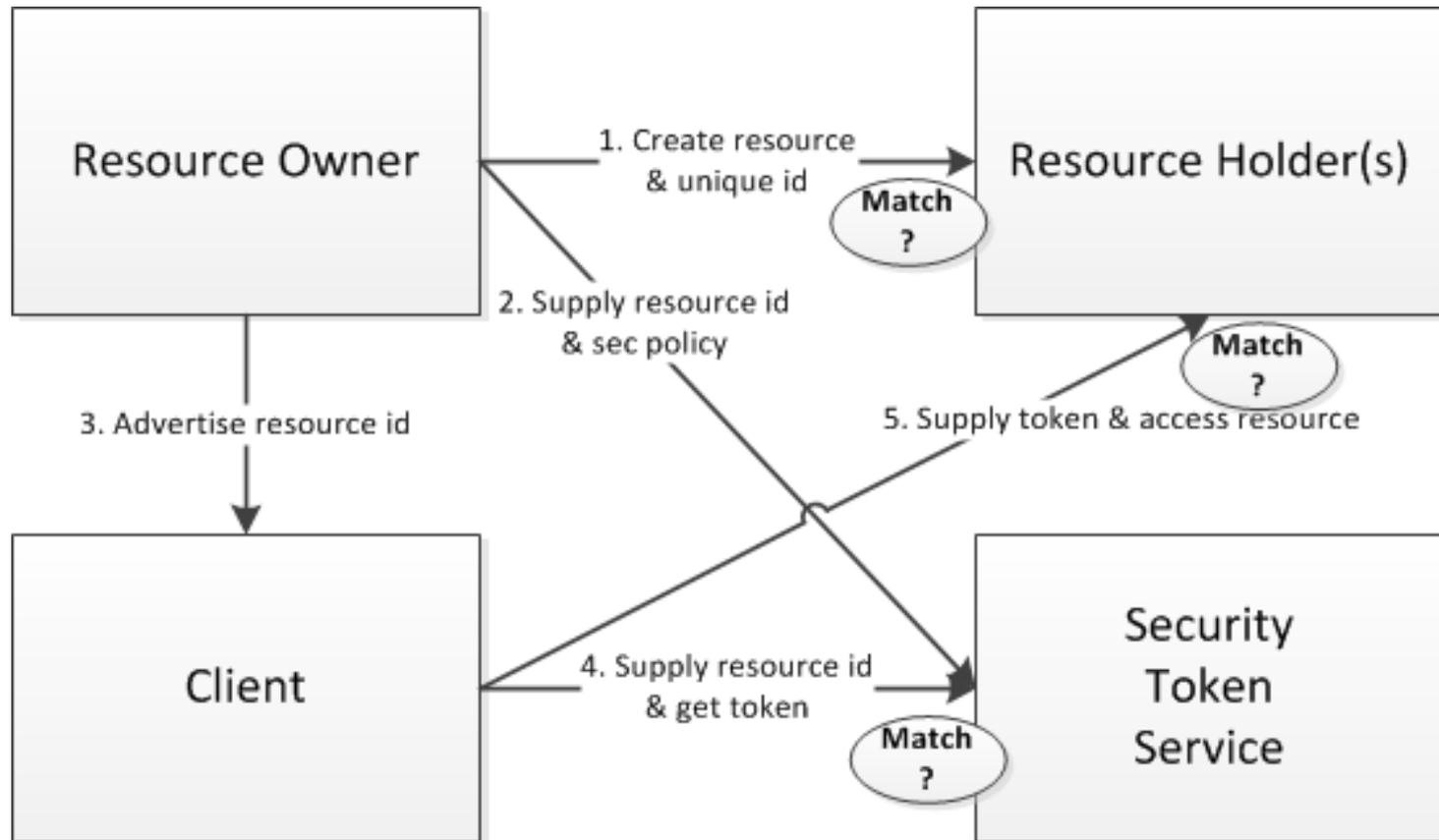
draft-iab-identifier-comparison-00

Dave Thaler
dthaler@microsoft.com

Issues in Identifier Comparison for Security Purposes

- Identifiers are often compared for security purposes, e.g.:
 - **Generation:**
 - Create a “unique” value that is “different” from previously generated ids
 - **Authentication:**
 - Match a security principal id to get keying material
 - Match keying material
 - **Authorization:**
 - Match a resource name to get ACL
 - Match a security principal id in ACL

Example of a Simple Security Exchange



Types of Identifiers

- Absolute: exact comparison
 - Ex: (binary) IPv4 address
- Definite: single globally-agreed on comparison
 - Ex: URI scheme name is ASCII-only case-insensitive and contains no %-escapes
- Indefinite: no single globally-agreed on algo.
 - Ex: human name

It's probably worse than you think...

Many identifiers are at best Definite and often turn out to be Indefinite.

Example: IPv4 literals or not? And do these match or not?

- 192.168.1.2
- 192.168.258
- 0xC0.0xA8.0x1.0x2
- 030052000402

Answer for all of the above: Maybe.

Even the term “standard dotted decimal” is ambiguous.

Effect of False Positives/Negatives

	“Grant on match”	“Deny on match”
False positive “match”	Elevation of Privilege	Denial of Service
False negative	Denial of Service	Elevation of Privilege

- EoP almost always far worse than DoS
 - E.g. RFC 3986 for URIs "comparison methods are designed to minimize false negatives while strictly avoiding false positives".
- ***Using URIs in a "deny on match" system can thus be problematic.***

Strawman Recommendations (1/2)

- Any system using both grant-on-match AND deny-on-match should not use Indefinite identifiers (Absolute ids have least chance of bugs).
- Any new identifiers should specify an Absolute or Definite comparison algorithm.
- If extensibility is allowed then the comparison algorithm should remain invariant, so that unrecognized extensions can be compared.

Strawman Recommendations (2/2)

- Some issues (e.g. unrecognized extensions) can be mitigated by treating such ids as invalid (see RFC 3696).
- Security protocols designed for use with other protocols should either:
 - a) specify the comparison algorithm, and ONLY be used by protocols that use the same algorithm, or
 - b) Support “matching algorithm” agility and use the one indicated by the using protocols.
 - When a collection of protocols are used together this may still mean all need to use the same algorithm.

Discussion

- i18n-discuss@iab.org