Internet Draft
Document: draft-ietf-rescap-req-00.txt
December 9, 1999
Expires June 9, 2000

John Beck
Sun Microsystems

ResCap Requirements

Status of this memo

Abstract

A variety of resource identifiers have been widely deployed on the Internet as a means of identifying various resources, services, and destinations. However, a means of attaching a set of attributes or characteristics to a given resource identifier and subsequently assessing those attributes or characteristics has not been specified and deployed.

Two tasks are envisioned. The first task will be to define a general resolution protocol that will translate resource identifiers to a list of attributes. The second task will be to define an administrative model and update protocol that can be used to set up and maintain the information the resolution protocol accesses.

This document defines the requirements for these two protocols.

0. Discussion

This draft is being discussed on the ResCap mailing list at <rescap@cs.utk.edu>. Subscription requests can be sent to <rescap-request@cs.utk.edu> (send an email message with the

word "subscribe" in the body). More information on the mailing list along with an archive of back messages is available at <ftp://cs.utk.edu/pub/rescap>.

1. Introduction

An attribute is a named characteristic of a resource identifier with a value that is meaningful in the context in which it is used. An attribute may be requested from a ResCap server. An example of an attribute might be a content type that an email address is capable of receiving.

A capability is a named set of one or more attributes that is meaningful in the context in which it is evaluated. A capability may be requested from a ResCap server. An example of a capability would be the complete set of content types that an email address could receive.

A particularly important resolution service of the general type described in the Abstract is one which, when given a mail address identifying a particular mail recipient, will return a series of attributes describing the capabilities of that recipient. This differs from a directory service in that no searching or other advanced query operations are involved. Likewise; this is not a discovery protocol.

2. General requirements

2.1 Security

The protocols must include support for authenticated messages between clients and servers. Specifically, it must be possible for a rescap client requesting information to reliably identify the server who is the source of the response. The client should be able to require a server to authenticate itself. The question as to whether the server is authorized to respond with the information requested is outside the scope of the protocols. See the Security Considerations section for more information.

Similarly, it must be possible for a rescap server to reliably identify the client who is making the request for information. The server should be able to require the client to authenticate itself. The question as to whether the client is authorized to request the information is outside the scope of the protocols. See the Security Considerations section for more information.

Because the principal purpose of the protocols is the dissemination of public information, support for confidentiality services is explicitly not required from the protocols. Nevertheless, specific attributes or capabilities may themselves be encrypted, unbeknownst to the protocol. Also, local security

policies may require the use of a secure transport layer that includes confidentiality services. See the Security Considerations section for more information.

3. Resolution protocol requirements

Throughout the rest of this section, "the protocol" refers to the resolution protocol.

3.1 Scalability

The protocol must be highly scalable both for number of entries in the database and number of entries per second resolved.

Example: Mail services with tens of millions of users could easily expect tens of millions of requests per day for client attribute information.

3.2 Reply data

The protocol must be able to support attributes and capabilities that are arbitrarily long text or binary values. The protocol should be optimized for the exchange of relatively short length resource identifiers and attribute values. By way of example, the distinction between short and long could be determined by whether or not the request and response fit in an ordinary UDP packet.

Servers must either provide the information requested, provide a referral indicating from where the information may be requested, or indicate the information is not available. Servers may forward requests on behalf of clients, but the forward must be transparent to the client.

3.3 Granularity

A mechanism needs to exist whereby a subset of capabilities for a resource can be fetched. I.e., the protocol request syntax should be able ask for one or more features instead of all of them at once. However, the client also needs to be able to ask for all capabilities known to the server without naming all of them.

Example: A client might only want to know the S/MIME capabilities of a recipient, but not care about its media features.

3.4 Expiration

Some means to indicate the expected lifetime of a capability is required, so that a client application can judge whether, or when, the information should be considered stale.

The expectation is that data will be infrequently updated, and that the granularity for time-stamps would be in seconds.

The protocol should also support a mechanism for indicating the "last changed date" of a given attribute.

Example: The ResCap server may believe that the recipient is only temporarily unable to receive large mail messages.

3.5 Referral

Some sort of request referral mechanism is needed. In other words, the protocol must support a mechanism whereby a response can indicate "I don't know, but go ask the ResCap server at address X." or "use the following URL to retrieve the ResCap response you requested." That is, the response might be a simple DNS name, or it might be a full ResCap URL.

Example: A server might delegate all requests for S/MIME certificate information to a different server that keeps track of that type of information.

3.6 Security

The protocol must be able to handle authenticated queries. The protocol must also support transmission of signed and/or encrypted responses. The protocol should allow for a ResCap server to hold and return "pre-signed" responses. This would allow it to hold capability information signed by the controller of the resource to which it refers, and also to sign responses off-line to avoid the need to perform signature computations at the time of responding to a query. This may imply a need to apply a signature to just part of a response.

Servers may require clients to use authenticated requests. Clients are not required to be able to create authenticated queries. Such clients will not be able to make requests of servers requiring authenticated queries; such servers might regard this as a feature.

Clients may require servers to use authenticated responses. Servers must be able to create authenticated responses.

Controls on which attributes will be announced should exist.

Note that consideration of transport-level security is out of scope here; an appropriate mechanism such as TLS or IPSec should be used instead.

Example: A server might give less information to a client that is unauthenticated than to one that is authenticated. Some

information from the server may be important enough for the server to want to prevent tampering, or even to prevent snoopers from reading.

3.7 Server location

DNS resource records should be used to determine a protocol server.

Example: The ResCap server that provides responses for resources at "example.com" might itself be running on a host other than "example.com", such as if the administrator of "example.com" has outsourced ResCap services to another company.

3.8 Preference

Preferences for a set of capabilities, if needed, are to be a supplied in a schema-specific fashion.

Example: A recipient might strongly prefer image/tiff files over image/jpeg because s/he can display image/tiff on his/her system without launching an external application.

3.9 Simplicity

The protocol should be sufficiently simple that it allows implementation of client and/or server functionality in very small, low cost devices (e.g. telephones, modems, printers, smart-cards, etc.).

3.10 Replication and Synchronization

We need to define the model for consistency between replicas - e.g. if a client is using one replica for queries and has to fail over to another, what assumptions can the client make about the consistency between the two replicas?

We should consider whether to assume strict master-slave as this choice would severely limit scalability of the service. Multi-master seems preferable, but this impacts the protocol as hooks are needed to allow clients to make sense of the data if they need to get it from multiple servers.

4. Administrative update protocol requirements

Throughout the rest of this section, "the protocol" refers to the administrative update protocol.

4.1 Access control

Authentication of anyone updating the database is required.

Example: Individual mail users should be able to update some or all of the information about them in the database, but such updates must be done with authentication to prevent others from maliciously entering false information.

4.2 Inheritance

The protocol must support inheritance. Specifically, mechanisms must be provided by which administrators can set default values for members of their administrative domains.

If a change is made to a default resource capability value that has previously been inherited by some other resource, then the inheritor should receive the new value.

Example: The media features for all addresses at a particular mail server might be the same because the mail server processes all messages at all addresses.

4.3 Scalability

The protocol must support atomic processing of request messages. This processing does not include updating any replicated sources for the information nor. The client should receive a completion response from the server, but the underlying protocol (if used over UDP) might require the client to retransmit its request at appropriate intervals until it receives such a response.

4.4 Security

The protocol must include support for authenticated messages. Servers and clients must use authenticated requests and responses to effect changes to attributes or capabilities.

4.5 Replication and Synchronization

SRV and/or NAPTR resource records may be used to determine a protocol server. [SRV, NAPTR]

5. Security Considerations

Security issues are discussed in sections 2.1, 3.6, 4.1 and 4.4 of this memo.

It is also worth noting that the data which these protocols will be designed to deal with are meant to be public, not private.

However, support for authentication should be included in the resolution protocol to permit administrators to restrict the attributes that are returned in response to a request according to a

locally specified security policy. The security policy may require the use of a transport security protocol, e.g., TLS [TLS], to provide additional security services not supported by these protocols.

6. Acknowledgements

The author would like to thank Paul Hoffman, Graham Klyne, Ned Freed, James Galvin and Keith Moore for their assistance.

7. References

[CONNEG] "A Syntax for Describing Media Feature Sets", RFC 2533.

[CONNEG-MEDIA] "MIME content types in media feature expressions", draft-ietf-conneg-feature-type.

[NAPTR] "Resolution of Uniform Resource Identifiers using the Domain Name System", RFC 2168.

[SRV] "A DNS RR for specifying the location of services (DNS SRV)", RFC 2052.

[TLS] "The TLS Protocol Version 1.0", RFC 2246

[UDP] "User Datagram Protocol", RFC 768.

8. Author's Address

John Beck
Sun Microsystems
901 San Antonio Road
M/S U-MPk-17-202
Palo Alto, CA 94303-4900
(650) 786-8078
jbeck+rescap@eng.sun.com

I. Issues

1. Does section 3.6 need further clean-up, particularly the first paragraph?
2. Are sections 4.1 and 4.4 redundant?
3. Section 3.10 currently lays out no requirements, but asks some questions and raises some issues which need to be addressed in order for the proper requirements to be determined.

X. Revision history

X.0 Initial revision (beck-00): April 15, 1999

X.1 beck-00 -> beck-01: May 14, 1999

* Added note to Abstract about this not being a discovery protocol.
* Examples added throughout.
* Section 1.2 renamed from "Long replies" to "Reply data", and clarified.
* Clarified section 1.4, and added note about support for a "last changed date".
* Section 1.6 renamed from "Privacy" to Security, and clarified, largely by absorbing section 2.2 (Privacy). As a result, section 2.3 (Inheritance) renumbered to 2.2 .
* New section 4 (Acknowledgements) added, and old sections 4 (References) and 5 (Author's Address) renumbered to 5 and 6.

X.2 beck-01 -> beck-02: June 15, 1999

* New section 1.9 (Simplicity) added.

X.3 beck-02 -> beck-02a: November 23, 1999

* Added notes about infrequent updates and granularity of seconds to section 1.4 (Expiration).
* Added note about transport-level security being out of scope to section 1.6 (Security), and did some word-smithing.
* Section 1.7 (Server location): specific references to SRV and NAPTR replaced by general reference to DNS. Deleted corresponding references from section 5 (References). Also rewrote example in section 1.7 .
* Section 1.8 (Preference) changed to be schema-specific.
* Added note about the effect of changing a default resource to section 2.2 (Inheritance).
* Added note about public vs. private date to section 3 (Security Considerations).
* Added appendix X (Revision history).

beck-02a -> beck-03: December 1, 1999

* New section 1 (Introduction) and 2 (General requirements); old sections 1 through 6 renumbered to 3 through 8.
* New section 4.3 (Scalability) added.
* Section 3.2 (Reply data) rewritten.

beck-03 -> rescap-00: December 9, 1999

* ResCap is now a WG, so document name changes.
* Added note about authentication to Security Considerations.
* Added appendix I for open issues.
* Added sections 2.1 and 4.4, and added to 3.6 (all on Security).
* Added sections 3.10 and 4.5 (on Replication and Synchronization).