REPUTE Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 2, 2014

N. Borenstein
Mimecast
M. Kucherawy

August 29, 2013

A Media Type for Reputation Interchange
draft-ietf-repute-media-type-11

Abstract

This document defines the format of reputation response data ("reputons"), the media-type for packaging it, and definition of a registry for the names of reputation applications and response sets.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 2, 2014.

Table of Contents

1. Introduction

This document defines a media type for use when answering a reputation query via the query method described in [I-D.REPUTE-QUERY-HTTP], which uses [HTTP].

Also included is the specification for an IANA registry to contain definitions and symbolic names for known reputation applications and corresponding response sets.

2. Terminology and Definitions

This section defines terms used in the rest of the document.

2.1. Reputon

A "reputon" is a single independent object containing reputation information. A particular query about a subject of interest will receive one or more reputons in response, depending on the nature of the data collected and reported by the server.

2.2. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

2.3. Other Definitions

Other terms of importance in this document are defined in [I-D.REPUTE-MODEL], the base document in this document series.

3. Description

The meta-format selected for the representaton of a reputon is Javascript Object Notation (JSON), defined in [JSON]. Accordingly, a new media type, "application/reputon+json", is defined for the JSON representation of reputational data, typically in response to a client making a request for such data about some subject. This media type has one optional parameter, "context", which names the semantic context (i.e., the specific sphere of reputation evaluation, or application) in which context the query is made and the response returned. If the parameter is absent, a generic reputation query (i.e., one not associated with a specific reputation application) is assumed for which only a simple reply is allowed.

The body of the media type consists of a JSON document that contains the reputation information requested. A detailed description of the expected structure of the reply is provided below.

3.1. Reputon Attributes

The key pieces of data found in a reputon for all reputation applications are defined as follows:

rater: The identity of the entity providing the reputation information, typically expressed as a DNS domain name.

assertion: A keyword indicating the specific assertion or claim being rated. In the absence of an "context" parameter on the media type, the reputon can only indicate generic goodness, with the default assertion "is-good", but each application is expected to define additional assertions.

rated: The identity of the entity being rated. The nature of this field is application-specific; it could be domain names, email addresses, driver's license numbers, or anything that uniquely identifies the entity being rated. Documents that define specific reputation applications are required to define syntax and semantics for this field.

rating: The overall rating score for that entity, expressed as a floating-point number between 0.0 and 1.0 inclusive. See Section 4 for discussion.

The following are OPTIONAL for all applications, to be used in contexts where they are appropriate:

confidence: the level of certainty the reputation provider has that the value presented is appropriate, expressed as a floating-point number between 0.0 and 1.0 inclusive.

normal-rating: An indication of what the reputation provider would normally expect as a rating for the subject. This allows the client to note that the current rating is or is not in line with expectations.

sample-size: The number of data points used to compute the rating, possibly an approximation. Expressed as an unsigned 64-bit integer. Consumers can assume that the count refers to distinct data points rather than a count of aggregations (for example, individual votes rather than aggregated vote counts) unless it is specified out-of-band that some other interpretation is more appropriate. The units are deliberately not normatively

specified, since not all reputation service providers will collect data the same way.

generated: A timestamp indicating when this value was generated. Expressed as the number of seconds since January 1, 1970 00:00 UTC.

expires: A timestamp indicating a time beyond which the score reported is likely not to be valid. Expressed as the number of seconds since January 1, 1970 00:00 UTC. See Section 5 for discussion.

A particular application that registers itself with IANA (per Section 7.2, below) can define additional application-specific attribute/value pairs beyond these standard ones.

An application service provider might operate with an enhanced form of common services, which might in turn prompt development and reporting of specialized reputation information. The details of the enhancements and specialized information are beyond the scope of this document, except that the underlying JSON syntax is extensible for encoding such provider-specific information.

4. Ratings

The score presented as the value in the rating attribute appears as a floating point value between 0.0 and 1.0 inclusive. The intent is that the definition of an assertion within an application will declare what the anchor values 0.0 and 1.0 specifically mean. Generally speaking, 1.0 implies full agreement with the assertion, while 0.0 indicates no support for the assertion.

The definition will also specify the type of scale in use when generating scores, to which all reputation service providers for that application space must adhere. This will allow a client to change which reputation service provider is being queried for a given without having to learn through some out-of-band method what the new provider's values mean. For example, a registration might state that ratings are linear, which would mean a score of "x" is twice as strong as a value of "x/2". It also allows easier aggregation of ratings collected from multiple reputation service providers.

5. Caching

A reputon can contain an "expires" field indicating a timestamp after which the client SHOULD NOT use the rating it contains, and SHOULD

issue a new query.

This specificaiton does not mandate any caching of ratings on the part of the client, but there are obvious operational benefits to doing so. In the context of reputation, a cached (and hence, stale) rating can cause desirable traffic to be identified as undesirable, or vice-versa.

Reputation data is typically most volatile when the subject of the reputation is young. Accordingly, if a service chooses to include expiration timestamps as part a reply, these values SHOULD be lower for subjects about which little data has been collected.

6. Reputon Structure

6.1. Syntax

A reputon expressed in JSON consists of an object that itself contains zero or more objects whose names are "reputon". Each reputon object is a set of key-value pairs, where the keys are the names of particular attributes that comprise a reputon (as listed above, or as provided with specific applications), and values are the content associated with those keys. The set of keys that make up a reputon within a given application are known as that application's "response set".

A reputon object typically contains a reply corresponding to the assertion for which a client made a specific request. For example, a client asking for assertion "sends-spam" about domain "example.com" would expect a reply consisting of a reputon making a "sends-spam" assertion about "example.com" and nothing more. If a client makes a request about a subject but does not specify an assertion of interest, the server can return reputons about any assertion for which it has data; in effect, the client has asked for any available information about the subject. A client that receives an irrelevant reputon simply ignores it.

An empty reputon is an acknowledgement by the server that the request has been received, and serves as a positive indication that the server does not have the information requested. This is semantically equivalent to returning a reputon with a "sample-size" of zero.

6.1.1. Formal Definition

Using [ABNF], the syntax of a reputon is:

```
reputon = "{" [ reputon-object
    *(value-separator reputon-object) ] "}"

reputon-object = "reputon" name-separator response-set

response-set = "{" reputon-element
    *(value-separator reputon-element) "}

reputon-element = rater-value / assertion-value / rated-value
                  / rating-value / conf-value / auth-value
                  / sample-value / gen-value / expire-value
                  / ext-value
    ; these can appear in any order, but MUST appear at most
    ; once each

rater-value = %x22 "rater" %x22 name-separator string

assertion-value = %x22 "assertion" %x22 name-separator string

rated-value = %x22 "rated" %x22 name-separator string

rating-value = %x22 "rating" %x22 name-separator number
    ; "number" MUST be between 0.0 and 1.0 inclusive

conf-value = %x22 "confidence" %x22 name-separator number
    ; "number" MUST be between 0.0 and 1.0 inclusive

auth-value = %x22 "rater-authenticity" %x22 name-separator number
    ; "number" MUST be between 0.0 and 1.0 inclusive

sample-value = %x22 "sample-size" %x22 name-separator int
    ; "int" MUST NOT be negative

gen-value = %x22 "generated" %x22 name-separator int
    ; "int" MUST NOT be negative

expire-value = %x22 "expires" %x22 name-separator int
    ; "int" MUST NOT be negative

ext-value = member
    ; specific syntax is specified in specific application
    ; registrations
```

The tokens "value-separator", "name-separator", "string", "int", and "member" are defined in [JSON].

6.2. Examples

The following simple example:

```
  Content-type: application/reputon+json

  {
    "reputon":
    {
      "rater": "RatingsRUs.example.com",
      "rater-authenticity": 1.0,
      "assertion": "is-good",
      "rated": "Alex Rodriguez",
      "rating": 0.99,
      "sample-size": 50000
    }
  }
```

...indicates we are absolutely sure (1.0) that the entity "RatingsRUs.example.com" consolidated 50000 data points (perhaps from everyone in Yankee Stadium) and concluded that Alex Rodriguez is very very good (0.99) at something. It doesn't tell us what he's good at, and while it might be playing baseball, it could just as well be paying his taxes on time.

A more sophisticated usage would define a baseball application with a response set of specific assertions, so that this example:

```
  Content-type: application/reputon+json; context="baseball"

  {
    "reputon":
    {
      "rater": "baseball-reference.example.com",
      "rater-authenticity": 1.0,
      "assertion": "hits-for-power",
      "rated": "Alex Rodriguez",
      "rating": 0.99,
      "sample-size": 50000
    }
  }
```

...would indicate that 50000 fans polled by the entity baseball-reference.example.com rate Alex Rodriguez very highly in hitting for power, whereas this example:

```
Content-type: application/reputon+json; context="baseball"

{
  "reputon":
  {
    "rater": "baseball-reference.example.com",
    "rater-authenticity": 1.0,
    "assertion": "clutch-hitter",
    "rated": "Alex Rodriguez",
    "rating": 0.4,
    "confidence": 0.2,
    "sample-size": 50000
  }
}
```

...would indicate that a similar poll indicated a somewhat weak consensus that Alex Rodriguez tends to choke in critical baseball situations.

In practice, it is expected that most reputons will have an "context" parameter to target an application-specific set of assertions.

The following is an example reputon generated using this schema, including the media type definition line that idenfities a specific reputation application context. Here, reputation agent "rep.example.net" is asserting within the context of the "email-id" application (see [I-D.REPUTE-EMAIL-IDENTIFIERS]) that "example.com" appears to be associated with spam 1.2% of the time, based on just short of 17 million messages analyzed or reported to date. The "email-id" application has declared the extension key "identity" to indicate how the subject identifier was used in the observed data, establishing some more specific semantics for the "rating" value. In this case, the extension is used to show the identity "example.com", the subject of the query, is extracted from the analyzed messages using the [DKIM] "d=" parameter for messages where signatures validate. The reputation agent is 95% confident of this result. (See [I-D.REPUTE-EMAIL-IDENTIFIERS] for details about the registered email identifiers application.)

```
Content-Type: application/reputon+json; context="email-id"

{
  "reputon":
  {
    "rater": "rep.example.net",
    "rater-authenticity": 0.95,
    "assertion": "spam",
    "identity": "dkim",
    "rated": "example.com",
    "rating": 0.012,
    "sample-size": 16938213,
    "updated": 1317795852
  }
}
```

7. IANA Considerations

This document presents two actions for IANA, namely the creation of the new media type "application/reputon+json" and the creation of a registry for reputation application types. Another document in this series creates an initial registry entry for the latter.

7.1. application/reputon+json Media Type Registration

This section provides the media type registration application from [MIME-REG] for processing by IANA:

To: media-types@iana.org

Subject: Registration of media type application/reputon+json

Type name: application

Subtype name: reputon+json

Required parameters: none

Optional parameters:

context: Names the reputation application in use within the reputon, which defines the valid assertions and any extensions that may also be valid (i.e., the response set) for that application. These are registered with IANA in the Reputation Application Registry as described in Section 7.2.

Encoding considerations: "7bit" encoding is sufficient and is used to maintain readability when viewed by non-MIME mail readers.

Security considerations: See Section 8 of [this document].

Interoperability considerations: Implementers may encounter "app" values, attribute/value pairs, or response set items that they do not support, which are to be ignored.

Published specification: [this document]

Applications that use this media type: Any application that wishes to query a service that provides reputation data using the "long form" defined in [I-D.REPUTE-QUERY-HTTP]. The example application is one that provides reputation expressions about DNS domain names found in email messages.

Additional information: The value of the "app" parameter is registered with IANA.

Person and email address to contact for further information:

Nathaniel Borenstein <nps@guppylake.com>

Murray S. Kucherawy <msk@cloudmark.com>

Intended usage: COMMON

Author:

Nathaniel Borenstein

Murray S. Kucherawy

Change controller: IESG

7.2. Reputation Applications Registry

IANA is requested to create the "Reputation Applications" registry. This registry will contain names of applications used with the application/reputon+json media type (and other media types that carry reputons), as defined by this document.

New registrations or updates are published in accordance with the "Specification Required" guidelines as described in [IANA-CONSIDERATIONS].

New registrations and updates are to contain the following

information:

1. Name of the application being registered or updated

2. Short description of the application (i.e., the class of entity about which it reports reputation data)

3. The document in which the application is defined

4. New or updated status, which is to be one of:

   current: The application is in current use

   deprecated: The application is in current use but its use is discouraged

   historic: The application is no longer in current use

5. A description of the subject of a query within this reputation, and a legal syntax for the same

6. An optional table of query parameters that are specific to this application; each table entry must include:

   Name: Name of the query parameter

   Status: (as above)

   Description: A short description of the purpose of this parameter

   Syntax: A reference to a description of valid syntax for the parameter's value

   Required: "yes" if the parameter is mandatory, "no" otherwise

7. A list of one or more assertions registered within this application; each table entry is to include:

   Name: Name of the assertion

   Description: A short description of the assertion, with specific meanings for values of 0.0 and 1.0

   Scale: A short description of the scale used in computing the value (see Section 4 of this document)

8. An optional list of one or more response set extension keys for use within this application; each table entry is to include:

   Name: Name of the extension key

   Description: A short description of the key's intended meaning

   Syntax: A description of valid values that can appear associated with the key

The names of attributes registered should be prefixed by the name of the application itself (e.g., the "foo" application registering a "bar" attribute should call it "foo-bar") to avoid namespace collisions.

8. Security Considerations

This document is primarily an IANA action registering a media type. It does not describe a new protocol that might introduce security considerations.

Discussion of the security and operational impacts of using reputation services in general can be found throughout [I-D.REPUTE-CONSIDERATIONS].

9. References

9.1. Normative References

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[I-D.REPUTE-MODEL] Borenstein, N. and M. Kucherawy, "A Model for Reputation Interchange", draft-ietf-repute-model (work in progress), November 2012.

[I-D.REPUTE-QUERY-HTTP] Borenstein, N. and M. Kucherawy, "Reputation Data Interchange using HTTP and XML", draft-ietf-repute-query-http (work in progress), November 2012.

[JSON] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[KEYWORDS]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[DKIM] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", RFC 6376, September 2011.

[HTTP] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[I-D.REPUTE-CONSIDERATIONS]
Kucherawy, M., "Operational Considerations Regarding Reputation Services", draft-ietf-repute-considerations (work in progress), November 2012.

[I-D.REPUTE-EMAIL-IDENTIFIERS]
Borenstein, N. and M. Kucherawy, "A Reputation Vocabulary for Email Identifiers", draft-ietf-repute-email-identifiers (work in progress), November 2012.

[IANA-CONSIDERATIONS]
Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.

[MIME-REG]
Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, December 2005.

Appendix A. Acknowledgments

The authors wish to acknowledge the contributions of the following to this specification: Frank Ellermann, Tony Hansen, Jeff Hodges, Simon Hunt, John Levine, David F. Skoll, and Mykyta Yevstifeyev.

Appendix B. Public Discussion

Public discussion of this suite of documents takes place on the domainrep@ietf.org mailing list. See https://www.ietf.org/mailman/listinfo/domainrep.

Authors' Addresses

Nathaniel Borenstein
Mimecast
203 Crescent St., Suite 303
Waltham, MA 02453
USA

Phone: +1 781 996 5340
Email: nsb@guppylake.com

Murray S. Kucherawy
270 Upland Drive
San Francisco, CA 94127
USA

Email: superuser@gmail.com