Internet Engineering Task Force (IETF)
Request for Comments: 7070
Category: Standards Track
ISSN: 2070-1721
November 2013

N. Borenstein
Mimecast
M. Kucherawy

An Architecture for Reputation Reporting

Abstract

This document describes a general architecture for a reputation-based service, allowing one to request reputation-related data over the Internet, where "reputation" refers to predictions or expectations about an entity or an identifier such as a domain name. The document roughly follows the recommendations of RFC 4101 for describing a protocol model.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Historically, many Internet protocols have operated between unauthenticated entities. For example, an email message’s author field (From:) [MAIL] can contain any display name or address and is not verified by the recipient or other agents along the delivery path. Similarly, a server that sends email using the Simple Mail Transfer Protocol [SMTP] trusts that the Domain Name System [DNS] has led it to the intended receiving server. Both kinds of trust are easily betrayed, opening the operation to subversion of some kind, which makes spam, phishing, and other attacks even easier than they would otherwise be.

In recent years, explicit identity authentication mechanisms have begun to see wider deployment. For example, the DomainKeys Identified Mail [DKIM] protocol permits associating a validated identifier to a message. This association is cryptographically strong, and is an improvement over the prior state of affairs, but it does not distinguish between identifiers of good actors and bad. Even when it is possible to validate the domain name in an author field (e.g., "trustworthy.example.com" in "john.doe@trustworthy.example.com"), there is no basis for knowing whether it is associated with a good actor who is worthy of trust. As a practical matter, both bad actors and good adopt basic authentication mechanisms like DKIM. In fact, bad actors tend to adopt them even more rapidly than the good actors do in the hope that some receivers will confuse identity authentication with identity assessment. The former merely means that the name is being used by its owner or their agent, while the latter makes a statement about the quality of the owner.

With the advent of these authentication protocols, it is possible to satisfy the requirement for a mechanism by which mutually trusted parties can exchange assessment information about other actors. For these purposes, we may usefully define "reputation" as "the estimation in which an identifiable actor is held, especially by the community or the Internet public generally". (This is based on the definition of "reputation" in [RANDOMHOUSE].) We may call an aggregation of individual assessments "reputation input".

While the need for reputation services has been perhaps especially clear in the email world, where abuses are commonplace, other Internet services are coming under attack and may have a similar need. For instance, a reputation mechanism could be useful in rating the security of web sites, the quality of service of an Internet Service Provider (ISP), or an Application Service Provider (ASP). More generally, there are many different opportunities for use of reputation services, such as customer satisfaction at e-commerce sites, and even things unrelated to Internet protocols, such as plumbers, hotels, or books. Just as human beings traditionally rely on the recommendations of trusted parties in the physical world, so too they can be expected to make use of such reputation services in a variety of applications on the Internet.

A full trust architecture encompasses a range of actors and activities, to enable an end-to-end service for creating, exchanging, and consuming trust-related information. One component of that is a query mechanism, to permit retrieval of a reputation. Not all such reputation services will need to convey the same information. Some need only to produce a basic rating, while others need to provide underlying detail. This is akin to the difference between check approval and a credit report.

An overall reckoning of goodness versus badness can be defined generically, but specific applications are likely to want to describe reputations for multiple attributes: an e-commerce site might be rated on price, speed of delivery, customer service, etc., and might receive very different ratings on each. Therefore, the architecture defines a generic query mechanism and basic format for reputation retrieval, but allows extensions for each application.

Omitted from this architecture is the means by which a reputation-reporting agent goes about collecting such data and the method for creating an evaluation. The mechanism defined here merely enables asking a question and getting an answer; the remainder of an overall service provided by such a reputation agent is specific to the implementation of that service and is out of scope here.

2. Overview

The basic premise of this reputation system involves a client that is seeking to evaluate content based on an identifier associated with the content, and a reputation service provider that collects, aggregates, and makes available for consumption, scores based on the collected data. Typically, client and service operators enter into some kind of agreement during which some parameters are exchanged, such as: the location at which the reputation service can be reached, the nature of the reputation data being offered, possibly some client authentication details, and the like.

Upon receipt of some content the client operator wishes to evaluate (an Internet message, for example), the client extracts from the content one or more identifiers of interest to be evaluated. Examples of this include the domain name found in the From: field of a message, or the domain name extracted from a valid DKIM signature.

Next, the goal is to ask the reputation service provider what the reputation of the extracted identifier is. The query will contain the identifier to be evaluated and possibly some context-specific information (such as to establish the context of the query, e.g., an email message) or client-specific information. The client typically folds the data in the response into whatever local evaluation logic it applies to decide what disposition the content deserves.

3. Related Documents

This document presents a high-level view of the reputation architecture.

For the purposes of sending and receiving reputation information, [RFC7071] defines a media type for containing responses to reputation queries, and a serialization format for these data (with examples). It also creates the registry for specific reputation contexts and the parameters related to them.

[RFC7072] describes how to construct and issue reputation queries and replies in the context of this architecture using the HyperText Transport Protocol (HTTP) as the query protocol.

Finally, [RFC7073] defines (and registers) a first, common, reputation application, namely the evaluation of portions of an email message as subjects for reputation queries and replies.

4. High-Level Architecture

This document outlines the reputation query and response mechanism. It provides the following definitions:

- Vocabulary for the current work and work of this type;

- The types and content of queries that can be supported;

- The extensible range of response information that can be provided;

- Query/response transport conventions.

It provides an extremely simple query/response model that can be carried over a variety of transports, including the Domain Name System. (Although not typically thought of as a 'transport', the DNS provides generic capabilities and can be thought of as a mechanism for transporting queries and responses that have nothing to do with Internet addresses, such as is done with a DNS BlockList [DNSBL].) Each specification for Repute transport is independent of any other specification.

The precise syntaxes of both the query and response are application specific. An application within this architecture defines the parameters available to queries of that type, and it also defines the data returned in response to any query.

4.1. Example of a Reputation Service Being Used

A reputation mechanism functions as a component of an overall service. A current example is that of an email system that uses DKIM [DKIM] to affix a stable identifier to a message and then uses that as a basis for evaluation:

```
        +-------------+                           +------------+
        |   Sender    |                           | Recipient  |
        +-------------+                           +------------+
               |                                         ^
               V                                         |
        +-------------+                           +------------+
        |     MSA     |                           |     MDA    |
        +-------------+                           +------------+
               |                                         ^
               |                                         |
               |                                  +------------+
               |                                  |  Handling  |
               |                                  |   Filter   |
               |                                  +------------+
               |                                         ^
               |                                         |
               |             +------------+       +------------+
               |             | Reputation |<=====>| Identifier |
               |             |  Service   |       |  Assessor  |
               |             +------------+       +------------+
               |                                         ^
               V                                         |
        +------------+  Responsible Identifier    +------------+
        | Identifier |. . . . . . . . . . . . . .>| Identifier |
        |   Signer   |         (DKIM)             |  Verifier  |
        +------------+                            +------------+
               |                                         ^
               V                                         |
        +-------------+       /~~~~~~~~~~\        +------+-----+
        |     MTA     |----->( other MTAs )------>|    MTA     |
        +-------------+       \~~~~~~~~~~/        +------------+
```

Figure 1: Actors in a Trust Sequence Using DKIM

See [EMAIL-ARCH] for a general description of the Internet messaging architecture. In particular, the terms Message Submission Agent (MSA), Message Delivery Agent (MDA), and Message Transfer Agent (MTA) are described there.

In this figure, the solid lines indicate the flow of a message; the dotted line indicates transfer of validated identifiers within the message content; and the double line shows the query and response of the reputation information.

Here, the DKIM Service provides one or more stable identifiers that is the basis for the reputation query. On receipt of a message from an MTA, the DKIM Service provides a (possibly empty) set of validated identifiers -- domain names, in this case -- that are the subjects of reputation queries made by the Identifier Assessor. The Identifier Assessor queries a Reputation Service to determine the reputation of the provided identifiers, and delivers the identifiers and their reputations to the Handling Filter. The Handling Filter makes a decision about whether and how to deliver the message to the recipient based on these and other inputs about the message, possibly including evaluation mechanisms in addition to DKIM.

5. Terminology and Definitions

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

5.1. Application

An "Application" is a specific context in which reputation queries are made. Some obvious popular examples include restaurants, movies, or providers of various services.

Applications have different sets of attributes of interest, and so the subjects of queries and the resulting responses will vary in order to describe the reputations of entities in their respective contexts. For example, the Application "movies" would have a different set of properties of interest and associated ratings (see below) from "restaurants". It is therefore necessary for them to be formally defined.

5.2. Response Set

A "Response Set" is a representation for data that are returned in response to a reputation query about a particular entity within the context of an Application. A Response Set will always contain at least the following components:

o the name of the entity being rated;

- the Assertion (see Section 5.3);

- the Rating (see Section 5.3).

The full content of the Response Set is specific to the Application; though all Applications have these few key Response Set fields in common, some of the reputation data returned in the evaluation of email senders would be different than that returned about a movie, restaurant, or baseball player. The specific meaning of a Rating is also specific to an Application.

A Response Set is declared in a specification document, along with a symbolic name representing the Application. The specifying documents will include the details of query parameters and responses particular to that Application. The symbolic names and corresponding specifying documents are registered with IANA in the "Reputation Applications" registry in order to prevent name collisions and provide convenient references to the documents.

IANA registries are created in [RFC7071].

5.3. Assertions and Ratings

One of the key properties of a Response Set is called an "Assertion". Assertions are claims made about the subject of a reputation query. For example, one might assert that a particular restaurant serves good food. In the context of this architecture, the assertion would be "serves good food".

Assertions are coupled with a numeric value called a "Rating", which is an indication of how much the party generating the Response Set agrees with the assertion being made. Ratings are typically expressed as a floating point value between 0.0 and 1.0 inclusive, with the former indicating no support for the assertion and the latter indicating total agreement with the assertion.

The documents that define future applications will also specify the type of scale in use when generating ratings, 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 without having to learn through some out-of-band method what the new provider’s ratings 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.

5.4. Reputon

A "reputon" is an object that comprises the basic response to a reputation query. It contains the Response Set relevant to the subject of the query in a serialized form. Its specific encoding is left to documents that implement this architecture.

6. Information Represented in the Protocol

Regardless of the transport selected for the interchange, the basic information to be represented in the protocol is fairly simple, and normally includes at least the following data:

In the query:

- the subject of the query;

- the name of the reputation context ("Application"; see Section 5.1);

- optionally, name(s) of the specific reputation assertions of interest.

Different transports, or different reputation contexts, might need additional query parameters.

In the response:

- the identity of the entity providing the reputation information;

- the identity of the entity being rated;

- the application context for the query (e.g., email address evaluation);

- the overall rating score for that entity.

Beyond this, arbitrary amounts of additional information might be included for specific uses of the service. The entire collection of data found in the response is the Response Set for that application and is defined in specifying documents as described above.

For example, a specification might be needed for a reputation Response Set for an "email-sending-domain"; the Response Set might include information on how often spam was received from that domain.

[RFC7071] defines a media type and format for reputation data, and [RFC7072] describes a protocol for exchanging such data.

7. Information Flow in the Reputation Query Protocol

The basic Response Set could be wrapped into a new MIME media type [MIME] or a DNS Resource Record (RR), and transported accordingly. It also could be the integral payload of a purpose-built protocol. For a basic request/response scenario, one entity (the client) will ask a second entity (the server) for reputation data about a third entity (the subject), and the second entity will respond with those data.

An application might benefit from an extremely lightweight mechanism, supporting constrained queries and responses, while others might need to support larger and more complex responses.

8. Privacy Considerations

8.1. Data in Transit

Some reputation exchanges can be sensitive, and should not be shared publicly. A client making use of this framework is explicitly revealing that it is interested in particular subjects, and the server is revealing what its information sources have reported about those subjects (in the aggregate). In the email context, for example, a client is revealing from whom it receives email, and the server is revealing what it (based on its aggregated data) believes to be true about those subjects.

These can be sensitive things that need to be secured, particularly when a client is talking to a server outside of its own administrative domain. Furthermore, certain types of reputation information are typically perceived as more sensitive than others; movie ratings, for example, are much less damaging if leaked than a person’s credit rating.

For interchanges that are sensitive to such exposures, it is imperative to protect the information from unauthorized access and viewing, and possibly add the capability to do object-level integrity and origin verification. Not all transport options can be adequately secured in these ways. In particular, DNS queries and responses are entirely insecure. Services need to use a transport method that provides adequate security when privacy-sensitive data are involved.

The architecture described here neither suggests nor precludes any particular transport mechanism for the data. An HTTP mechanism is defined in [RFC7072], and email-based mechanisms are also envisioned. For HTTP, use of HTTP over Transport Layer Security [HTTP-TLS] is very strongly advised. For email, mechanisms such as OpenPGP [OPENPGP] and S/MIME [SMIME] are similarly advised.

8.2. Aggregation

The data that are collected as input to a reputation calculation are, in essence, a statement by one party about the actions or output of another. What one party says about another is often meant to be kept in confidence. Accordingly, steps often need to be taken to secure the submission of these input data to a reputation service provider.

Moreover, although the aggregated reputation is the product provided by this service, its inadvertent exposure can have undesirable effects. Just as the collection of data about a subject needs due consideration to privacy and security, so too does the output and storage of whatever aggregation the service provider applies.

8.3. Collection of Data

The basic notion of collection and storage of reputation data is obviously a privacy issue in that the opinions of one party about another are likely to be sensitive. Inadvertent or unauthorized exposure of those data can lead to personal or commercial damage.

8.4. Queries Can Reveal Information

When a client asks a service provider about a particular subject, the service provider can infer the existence of that subject and begin observing which clients ask about it. This can be an unanticipated leak of private information.

8.5. Compromised Relationships

Reputation services that limit queries to authorized clients can cause private information, such as the reputations themselves or the data used to compute them, to be revealed if the client credentials are compromised. It is critical to safeguard not only the interchange of reputation information, and the information once it has been delivered to the client, but the ability to issue requests for information as well.

An important consideration here is that compromised credentials are mainly an exposure of some third party (whose reputation is improperly revealed) rather than the client or the server.

9. Security Considerations

This document introduces an overall protocol architecture, but no implementation details. As such, the security considerations presented here are very high level. The detailed analysis of the various specific components of the protocol can be found in the documents that instantiate this architecture.

9.1. Biased Reputation Agents

As with [VBR], an agent seeking to make use of a reputation reporting service is placing some trust that the service presents an unbiased "opinion" of the object about which reputation is being returned. The result of trusting the data is, presumably, to guide action taken by the reputation client. It follows, then, that bias in the reputation service can adversely affect the client. Clients therefore need to be aware of this possibility and the effect it might have. For example, a biased system returning a reputation about a DNS domain found in email messages could result in the admission of spam, phishing, or malware through a mail gateway (by rating the domain name more favorably than warranted) or could result in the needless rejection or delay of mail (by rating the domain more unfavorably than warranted). As a possible mitigation strategy, clients might seek to interact only with reputation services that offer some disclosure of the computation methods for the results they return. Such disclosure and evaluation is beyond the scope of the present document.

Similarly, a client placing trust in the results returned by such a service might suffer if the service itself is compromised, returning biased results under the control of an attacker without the knowledge of the agency providing the reputation service. This might result from an attack on the data being returned at the source, or from a man-in-the-middle attack. Protocols, therefore, need to be designed so as to be as resilient against such attacks as possible.

9.2. Malformed Messages

Both clients and servers of reputation systems need to be resistant to attacks that involve malformed messages, deliberate or otherwise. Malformations can be used to confound clients and servers alike in terms of identifying the party or parties responsible for the content under evaluation. This can result in delivery of undesirable or even dangerous content.

9.3. Further Discussion

Involving a third party (in this case, a reputation service provider) that can influence the handling of incoming content involves ceding some amount of control to that third party. Numerous other topics related to the management, operation, and safe use of reputation systems can be found in [CONSIDERATIONS].

10. Informative References

[CONSIDERATIONS]
Kucherawy, M., "Operational Considerations Regarding Reputation Services", Work in Progress, May 2013.

[DKIM] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, September 2011.

[DNS] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[DNSBL] Levine, J., "DNS Blacklists and Whitelists", RFC 5782, February 2010.

[EMAIL-ARCH]
Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

[HTTP-TLS] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[MAIL] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[OPENPGP] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RANDOMHOUSE]
"Random House Webster’s Dictionary, Revised Edition", ISBN 978-0-345-44725-8, June 2001.

[RFC7071] Borenstein, N. and M. Kucherawy, "A Media Type for Reputation Interchange", RFC 7071, November 2013.

[RFC7072] Borenstein, N. and M. Kucherawy, "A Reputation Query Protocol", RFC 7072, November 2013.

[RFC7073] Borenstein, N. and M. Kucherawy, "A Reputation Response Set for Email Identifiers", RFC 7073, November 2013.

[SMIME] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[VBR] Hoffman, P., Levine, J., and A. Hathcock, "Vouch By Reference", RFC 5518, April 2009.

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