Network Working Group
Internet-Draft
Expires: April 3, 2004

A. Barbir
Nortel Networks

October 4, 2003

OPES processor and end points communications
draft-ietf-opes-end-comm-03

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 3, 2004.

Copyright Notice

Abstract

This memo documents tracing and non-blocking requirements for Open Pluggable Edge Services (OPES).

Table of Contents

1. Introduction

The Open Pluggable Edge Services (OPES) architecture [8] enables cooperative application services (OPES services) between a data provider, a data consumer, and zero or more OPES processors. The application services under consideration analyze and possibly transform application-level messages exchanged between the data provider and the data consumer.

This work specify the requirements for providing tracing functionality for the OPES architecture [8]. Tracing functionality enables a data provider application to detect inappropriate actions that are performed by OPES entities. The work also develops requirements that can be used to fulfill IAB Notification and Non-Blocking requirements [2].

The architecture document requires [8] that tracing be supported in-band. This design goal limits the type of application protocols that OPES can support. The details of what a trace record can convey is also dependent on the choice of the application level protocol.

For these reasons, this work documents requirements for application protocols that need to support OPES traces and non-blocking. However, the architecture does not prevent implementers of developing out-of-band protocols and techniques to address these limitation.

2. OPES System

This sections provides a definition of OPES System. This is needed in order to define what is traceable in an OPES Flow.

Definition: An OPES System is a set of all OPES entities authorized by either the data provider or the data consumer application to process a given application message.

The nature of the authorization agreement determines if authority delegation is transitive (meaning an authorized entities is authorized to include other entities). Transitive authority delegation is common in information systems.

If specific authority agreements allow for re-delegation, an OPES system can be formed by induction. In this case, an OPES system starts with entities directly authorized by a data provider (or a data consumer) application. The OPES system then includes any OPES entity authorized by an entity that is already in the OPES system. The authority delegation is always viewed in the context of a given application message.

An OPES System is defined on an application message basis. Having an authority to process a message does not imply being involved in message processing. Thus, some OPES system members may not participate in processing of a message. Similarly, some members may process the same message several times.

The above definition implies that there can be no more than one OPES system processing a single message at a given time. This is based on the assumption that there is a single data provider and a single data consumer as far as a given application message is concerned.

For example, consider a Content Delivery Network (CDN) delivering an image on behalf of a busy web site. OPES processors and services that the CDN uses to adapt and deliver the message comprise an OPES system. In a more complex example, an OPES System would contain CDN entries as well as 3rd-party OPES entities that CDN engages to perform adaptations (e.g., to adjust image quality).

3. Requirements for OPES Tracing

In an OPES System tracing is defined as the inclusion of necessary information within a message in an OPES Flow that identify the set of transformations or adaptations that have been performed on it before its delivery to an end point (for example, the data consumer application). An OPES trace represents a snap shot of the tracing information that have been added to a given application message.

In an OPES System tracing is performed on per message basis. Trace format is dependent on the application protocol that is being adapted by OPES. A Data consumer application can use OPES trace to infer the actions that have been performed by the OPES system. Actions are the set of authorized OPES services that were performed by OPES entities in an OPES System.

Providing tracing information, MUST take into account the following considerations:

- Providers may be hesitant to reveal information about their internal network infrastructure.
- Within a service provider network, OPES processors may be configured to use non-routable, private IP addresses.
- A Data consumer applications would prefer to have a single point of contact regarding the trace information.

3.1 What is traceable in an OPES Flow?

This section focuses on identifying traceable entities in an OPES Flow.

Tracing information provides a data consumer application (or a data provider application) with useful information without tracing the exact OPES Processor or callout servers that adapted the data. For example, some OPES services are message-agnostic and operate on message content or parts of a message. Such services cannot manipulate message headers. Hence, a data consumer application would be interested in knowing that a translation service on the message content was performed. It does not need to know the exact entity that has performed the service.

There are two distinct uses of OPES traces. First, a trace enables an "end (content provider or consumer) to detect the presence of OPES processors within an OPES System. Such "end" should be able to see a trace entry, but does not need to be able to interpret it beyond

identification of the OPES System.

Second, the OPES System administrator is expected to be able to interpret the contents of an OPES trace. The trace can be provided by an end (data consumer or provider) as an opaque string. The administrator can use the trace information to identify the participating OPES processor(s). The administrator can use the trace to identify the applied adaptation services along with other message-specific information.

Since the administrators of various OPES Systems can have various ways of looking into tracing, they MAY require the choice of freedom in what to put in trace records and how to format them. Trace records should be easy to extend beyond basic OPES requirements. Trace management algorithms should treat trace records as opaque data to the extent possible.

At the implementation level, for a given trace, an OPES entity involved in handling the corresponding application message is traceable or traced if information about it appears in that trace. OPES entities have different levels of traceability requirements. Specifically,

- An OPES system MUST add its entry to the trace.
- An OPES processor SHOULD add its entry to the trace.
- An OPES service May add its entry to the trace.
- An OPES entity MAY manage trace information from entities that are under its control. For example, an OPES processor may add or remove callout service entries in order to manage the size of a trace.

From an OPES context, a good tracing approach is similar to a trouble ticket ready for submission to a known address. The address is printed on the ticket. The trace in itself is not necessarily a detailed description of what has happened. It is the responsibility of the operator to resolve the problems.

3.2 Requirements for Information Related to Traceable Entities

The following MUST requirements apply for information as related to entities that are traceable in an OPES flow:

- Identification of the OPES System privacy policy at the time it dealt with the message.

- Identification of the party responsible for setting and enforcing that policy.
- Information pointing to a technical contact.
- Information that identifies, to the technical contact, the OPES processors involved in processing the message.

3.3 Requirements for OPES processors

The requirements for OPES processors that are applicable to tracing are:

- Each OPES processor MUST be uniquely identified in an OPES System.
- Each OPES processor MUST support tracing, policy can be used to turn tracing on and to determine its granularity.
- If tracing is turned on, then the OPES processor MUST add its identification to the trace.
- OPES processor SHOULD be able to trace it's own invocation and service(s) execution since it understands the application protocol. To fulfill this:
  - An OPES processor MAY have a fixed configuration that enable it to respond to tracing inquires. For example, entity X performs service Y and so on.
  - An OPES processor MAY package tracing information related to the entities that it control based on the policy of a given OPES System. For example, the trace may state that service W was performed. The OPES processor knows that service W is composed of services X, Y and Z in a given order
- An OPES processor SHOULD add to the trace identification of every callout service that processed the application message.
- An OPES processor MAY delegate trace management to a callout service within the same OPES System.

3.4 Requirements for callout servers

In an OPES system, it is the task of an OPES processor to add trace records to application messages. However, in some cases, callout servers May add trace information to application messages. This

should be done under the control of the OPES System provider.

3.5 Protocol Binding

The task of adding tracing information is application protocol specific. Separate documents will address HTTP and other protocols. This work documents what tracing information is required and some common tracing elements.

4. Non-Blocking

In [9] recommendation addresses the issue of non-blocking in an OPES System. The recommendation is restated below for ease of reference.

> (3.3) Non-blocking: If there exists a non-OPES version of content available from the content provider, the OPES architecture must not prevent users from retrieving this non-OPES version from the content provider.

The IAB recommendation implies that it is up to the content provider to make non-OPES versions of a given content available. The actual meaning of non-OPES version of the content depended on the agreement between the OPES provider and the content provider. The agreement can allow OPES to perform some services (such as logging services) and prevent it from performing other services (such as data to audio transformation).

Whether an OPES System honor a non-blocking request from a data consumer application (user) can also be a function of deployment. Consider the case where Company A has as contract with an OPES provider to perform virus checking on all e-mail attachments. An employee X of Company A can issue a non-blocking request for the virus scanning service. However, the request could be ignored by the OPES provider since it contradicts its agreement with Company A. As a second example, a user may issue a non-blocking request for adult content, this request may be declined by the OPES provider simply because it contradicts its internal policy or its agreement with the end subscriber.

In some cases, a data consumer application will issue a non-blocking request since it suspects that the OPES System is corrupting the data. For example, an OPES entity has determined that a Virus is present in an attachment, while the user is aware that some versions of virus scanners will make that mistake. In this case, the user can use the non-blocking technique (can be used in combination with the tracing facility) to solve the problem. However, whether the OPES System will honor the non-blocking request or not is still a function of the deployment scenario, content availability and related policies.

Like tracing, Non-blocking operates on per application message bases. Non-Blocking is an end-end operation as opposed to a hop-by-hop operation. Non-blocking requests are generally client centric and go in the opposite direction of tracing requests. Non-blocking can be performed out of band or in-band. This work requires non-blocking to be performed in-band as an extension to an application specific protocol. Non-OPES entities should be able to safely ignore the

extensions. The work does not prevent OPES Systems from developing their own out of band protocols.

Non-blocking format is dependent on the application protocol that is being adapted by OPES. For a given application protocol, in an OPES System there can be services that operate on application message headers and those that just operate on content. This mix of service requires that an OPES processor that is calling the service(s) to handle the non-blocking request. In some cases, the first OPES processor that will get the non-blocking request may not be the first OPES processor that will know whether a non-OPES version of the content is available or not.

In an OPES System, the OPES provider is expected to configure at least one OPES processor to process a non-blocking header based on content availability and related policies. In this case the OPES processor is expected to determine the set of services that will be bypassed (or those services that will be performed) or whether the request should be forwarded directly to the data provider application (origin content provider).

Although, IAB recommendation (3.3) has intended for non-blocking approach to be used as a vehicle to bypass faulty OPES intermediaries. However, this work recognizes that the same technique can be used to enable a data consumer application to select the set of services that it would like to be bypassed for a given application message. For this reason, a non-blocking request is viewed as a bypass instruction that contains a URI that identifies an OPES entity or a group of OPES entities that perform a service (or services) to be bypassed. An instruction may contain more than one such URI. A special wildcard identifier can be used to represent all possible URIs (i.e., all possible OPES services). This version of the work requires that all non-blocking instructions to use the wildcard approach.

For example, an application level protocol (such as HTTP) can be extended to include the following OPES non-blocking related header:

```
OPES-Bypass: *
```

The following requirements apply for non-blocking feature:

- An OPES System MUST support the non-blocking feature for requests of non-OPES content for a given application message.

- An OPES System MUST treat the non-blocking feature as an end-to-end operation.

    * This means that there MUST be at least one OPES processor in an OPES System that knows how to interpret and process the non-blocking feature.

    * The recipient MUST forward the bypass instructions to the next application hop provided that the next hop speaks application protocol with OPES bypass support.

    * This requirement applies to all bypass instructions, including those that identify known-to-recipient entities.

- Application-specific bindings MUST map the above non-blocking mechanism to their application protocol.

End users may not be able to know if their non-blocking request was honored or not by the OPES System. In this case, it would be beneficial if tracing can provide additional information regarding whether a non-blocking request was honored or not. For this reason, the following requirement also apply to the tracing facility:

- An OPES System SHOULD assist the data consumer application in determining if a non-blocking request was performed by the system.

Assistance is viewed as the addition of information about services that were skipped and those that could not be bypassed.

5. IANA considerations

This work does not require any IANA consideration since any actions will be addressed in [6].

6. Security Considerations

The security considerations for OPES are documented in [7]. This document is a requirement document for tracing and non-blocking and as such does not develop any new protocols that require security considerations.

Normative References

[1] A. Barbir et al., "OPES Use Cases and Deployment Scenarios", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-scenarios-01.txt, August 2002.

[2] Floyd, S. and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, January 2002.

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

[4] A. Barbir et al., "Policy, Authorization and Enforcement Requirements of OPES", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-authorization-02.txt, February 2003.

[5] Rousskov, A., "OPES Callout Protocol Core", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-ocp-core-01.txt, August 2003.

[6] Rousskov, A., "HTTP adaptation with OPES", Internet-Draft TBD, September 2003.

[7] A. Barbir et al., "Security Threats and Risks for Open Pluggable Edge Services", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-threats-02.txt, February 2003.

[8] A. Barbir et al., "An Architecture for Open Pluggable Edge Services (OPES)", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-architecture-04, December 2002.

[9] A. Barbir et al., "OPES Treatment of IAB Considerations", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-iab-01.txt, February 2004.

Informative References

[10] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

[11] L. Cranor, et. al, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", W3C Recommendation 16 http://www.w3.org/TR/2002/REC-P3P-20020416/ , April 2002.

Author's Address

Abbie Barbir
Nortel Networks
3500 Carling Avenue
Nepean, Ontario K2H 8E9
Canada

Phone: +1 613 763 5229
EMail: abbieb@nortelnetworks.com

Appendix A. Acknowledgements

Several people has contributed to this work. Many thanks to: Alex Rousskov, Hilarie Orman, Oscar Batuner, Markus Huffman, Martin Stecher, Marshall Rose and Reinaldo Penno.

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.