Network Working Group
Internet-Draft
Expires: December 10, 2003

A. Barbir
Nortel Networks

June 11, 2003

OPES processor and end points communications
draft-ietf-opes-end-comm-00

Status of this Memo

Copyright Notice

Abstract

This memo documents tracing 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.

The execution of such services is governed by a set of rules installed on the OPES processor. The rules enforcement can trigger the execution of service applications local to the OPES processor. Alternatively, the OPES processor can distribute the responsibility of service execution by communicating and collaborating with one or more remote callout servers. As described in [8], an OPES processor communicates with and invokes services on a callout server by using a callout protocol.

The work specify the requirements for providing tracing functionality for the OPES architecture [8]. This document specifies tracing mechanisms that the OPES architecture could provide that enable data provider application to detect inappropriate clinet centric actions by OPES entities. The work focus on developing tracing requirements that can be used to fulfil the notification and Non-Blocking requirements [2].

In the OPES architecture document [8], there is a requirement of relaying tracing information in-band. This work investigates this possibility and discusses possible methods that could be used to detect faulty OPES processors or callout servers by end points in an OPES flow.

The document is organized as follows: ....

2. OPES Tracing

Before discussing what is traceable in an OPES flow, it is beneficial to define what tracing means. Tracing is defined as the inclusion of necessary information within a message in an OPES flow that could be used to identify the set of transformations or adpatations that have been performed on its content before its delivery to an end point (the data consumer application).

- OPES trace: application message information about OPES entities that adapted that message

- OPES tracing: the process of including, manipulating, and interpreting an OPES trace

To emphasize, the above definition means that OPES tracing SHOULD be performed on per message basis. Trace format is dependent on the application protocol being adapted by OPES. Data consumer application can use OPES trace to infer the actions that have been performed by OPES system(s). The architecture document requires [8] that tracing be supported in-band.

2.1 What is traceable in an OPES Flow?

- The data consumer application end point MUST be able to identify the OPES processors that have acted on an application message.

- The data consumer application end point SHOULD be able to identify OPES services (including callout services) that were performed on request/responses that are part of an application message.

- TBD

- TBD

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 be traceable

- An OPES processor SHOULD be traceable

- An OPES service MAY be traceable

- Editor Note: Need to define an OPES System properly

2.2 Requirements for Information Related to Traceable Entities?

The requirements for information as related to entities that are terceable in an OPES flow are:

- The 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 messag

- TBD

3. Requirements for OPES systems

Editor Note: Need to define OPES System and state requirements

4. Requirements for OPES processors

TBD

5. Requirements for callout servers

If it is the task of an OPES processor to add trace records to application messages, then callout servers that uses the OCP protocol are not affected by tracing requirements.In order for an OCP protocol to be tracing neutral, the OPES server SHOULD be able to meet the following requirements:

- o Callout services adapt payload regardless of the application protocol in use and leave header adjustment to OPES processor.

- o OPES processor SHOULD be able to trace its own invocation and service(s) execution because OPES processor understand the application protocol.

- o Callout servers MAY be able to add their own OPES trace records to application level messages.

- o TBD

6. Privacy considerations

6.1 Tracing and Trust Domains

A trust domain may include several OPES systems and entities. Within a trust domain, there MUST be at least support for one trace entry per system. Entities outside of that system may or may not see any traces, depending on domain policies or configuration. For example, if an OPES system is on the content provider "side", end-users are not guaranteed any traces. If an OPES system is working inside end-user domain, the origin server is not guaranteed any traces related to user requests.

7. How to Support Tracing

In order to support tracing, the following aspects must be addressed:

- There MUST be a System Identifier that identify a domain that is employing an OPES system.

- An OPES processor MUST be able to be uniquely identified (MUST have an Identifier) within a system.

- An OPES processor MUST add its identification to the trace.

- An OPES processor SHOULD add to the trace identification of every callout service that received the application message.

- An OPES processor MUST add to the trace identification of the "system/entity" it belongs to. "System" ID MUST make it possible to access "system" privacy policy.

- An OPES processor MAY group the above information for sequential trace entries having the same "system/entity" ID. In other words, trace entries produced within the same "system/entity" MAY be merged/aggregated into a single less detailed trace entry.

- An OPES processor MAY delegate trace management to a callout service within the same "system/entity".

TBD

7.1 Tracing and OPES System Granularity

There are two distinct uses of traces. First, is to SHOULD enable the "end (content producer or consumer) to detect OPES processor presence within end's trust domain. Such "end" should be able to see a trace entry, but does not need to be able to interpret it beyond identification of the trust domain(s).

Second, the domain administrator SHOULD be able to take a trace entry (possibly supplied by an "end? as an opaque string) and interpret it. The administrator must be able to identify OPES processor(s) involved and may be able to identify applied adaptation services along with other message-specific information. That information SHOULD help to explain what OPES agent(s) were involved and what they did. It may be impractical to provide all the required information in all cases. This document view a trace record as a hint, as opposed to an exhaustive audit.

Since the administrators of various trust domains 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.

It is not expected that entities in one trust domain to be able to get all OPES-related feedback from entities in other trust domains. For example, if an end-user suspects that a served is corrupted by a callout service, there is no guarantee that the use will be able to identify that service, contact its owner, or debug it _unless_ the service is within my trust domain. This is no different from the current situation where it is impossible, in general, to know the contact person for an application on an origin server that generates corrupted HTML; and even if the person is known, one should not expect that person to respond to end-user queries.

7.2 Requirements for In-Band Tracing

The OPES architecture [8] states that traces must be in-band. The support of this design specification is dependent on the specifics of the message application level protocol that is being used in an OPES flow. In-band tracing 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, the work will document requirements for application protocols that need to support OPES traces. However, the architecture does not prevent implementers of developing out-of-band protocols and techniques to address the above limitation.

7.2.1 Tracing Information Granularity and Persistence levels Requirements

In order to be able to trace entities that have acted on an application message in an OPES flow, there may be requirements to keep information that is related to the following:

- Message-related informatio: All data that describes specific actions performed on the message SHOULD be provided with that message, as there is no other way to find message level details later.

- Session related information: Session level data MUST be preserved for the duration of the session. OPES processor is responsible for inserting notifications if session-level information changes.

- End-point related data: What profile is activated? Where to get

profile details? Where to set preferences?

- TBD

7.3 Protocol Binding

How tracing is added is application protocol-specific and will be documented in separate drafts. This work documents what tracing information is required and some common tracing elements.

7.4 Tracing scenarios and examples

TBD

8. IAB considerations

This section examines IAB [2] considerations (3.1) and (3.2) regarding notification in an OPES architecture. The IAB considerations are reiterated here for ease of reference.

Notification propagates in opposite direction of tracing and cannot be attached to application messages that it notifies about. Notification can be done out-band and may require the development of a new protocol. The direction of data flow for tracing and notification are deoicted in Figure 1.

```
                                      Notification
                +------------------------------------------------
                |                                               |
                |                                               V
          +---------------+            +-------+       +---------------+
          |               |            |       |       | Data Provider |
          | Data Consumer | Tracing    | OPES  |<----->|  Application  |
          |  Application  |<-----------|       |       +---------------+
          +---------------+            +-------+
                                           ^
                                           |OCP
                                           |
                                           V
                                       +---------+
                                       | Callout |
                                       | Server  |
                                       +---------+
```

Figure 1: Notification Flow

8.1 Notification Concerns

Notifications for every HTTP request can burden some content providers. Therefore, it might be preferable to consider mechanisms that allow for the explicit request of notification. Hence, a mechanism for explicit request of notification May be required.

Furthermore, end point privacy is a concern. An end user may consider information about OPES services applied on their behalf as private. For example, if translation for braille device has been applied, it can be concluded that the user is having eyesight problems; such

information may be misused if the user is applying for a job online. Similarly, a content provider may consider information about its OPES services private. For example, use of a specific OPES intermediary by a high traffic volume site may indicate business alliances that have not been publicly announced yet. Another example of privacy, include situations where a user may not want to reveal to any content provider all the OPES services that have been applied on their behalf. For example, why should every content provider know what exact virus scanner a user is using?

Security is also a concern. An attacker may benefit from knowledge of internal OPES services layout, execution order, software versions and other information that are likely to be present in automated notifications.

The level of available details in notifications versus content provider interest in supporting notification is a concern. Experience shows that content providers often require very detailed information about user actions to be interested in notifications at all. For example, Hit Metering protocol [11] has been designed to supply content providers with proxy cache hit counts, in an effort to reduce cache busting behavior which was caused by content providers desire to get accurate site "access counts". However, the Hit Metering protocol is currently not widely deployed. This is because the protocol does not supply content providers with information such as client IP addresses, browser versions, or cookies.

The Hit Metering experience is relevant because Hit Metering protocol was designed to do for HTTP caching intermediaries what OPES notifications are meant to do for OPES intermediaries. Thus, it is important to have the right balance when specifying the notofication requirements for OPES.

In this document, IAB choice of "Notification" label is interpreted as "Notification assistance" (i.e. making notifications meaningful) and is not be interpreted as a "Notification protocol". Therefore, the work treats IAB considerations (3.1 and 3.2) as informative (not normative).

8.1.1 Addressing IAB Consideration 3.1

The consideration is restated below for ease of reference.

(3.1) Notification: The overall OPES framework needs to assist content providers in detecting and responding to client-centric actions by OPES intermediaries that are deemed inappropriate by the content provider.

IAB consideration (3.1) suggests that the overall OPES framework needs to assist content providers in detecting and responding to client-centric actions by OPES intermediaries that are deemed inappropriate by the content provider.

It is important to note that most client-centric actions happen after the application message has left the content provider(s). Thus, notifications cannot be piggy-backed to application messages and have to travel in the opposite direction of traces, see Figure 1. To address this requirement directly, one would have to develop an out of band protocol to support notification.

At this stage, there is no need to develop an out of band protocol to support notification, since requiring the OPES architecture to having a tracing facility can fulfil the objectives of notification. In this regard, it is recommended that tracing MUST be always-on, just like HTTP Via headers. This should eliminate notification as a separate requirement.

8.1.2 Addressing IAB Consideration 3.2

The consideration is restated below for ease of reference.

(3.2) Notification: The overall OPES framework should assist end users in detecting the behavior of OPES intermediaries, potentially allowing them to identify imperfect or compromised intermediaries.

TBD

If the OPES end points cooperate then notification can be supported by tracing. Content providers that suspect or experience difficulties can do any of the following:

- Check whether requests they receive pass through OPES intermediaries. Presence of OPES tracing info will determine that. This check is only possible for request/response protocols. For other protocols (e.g., broadcast or push), the provider would have to assume that OPES intermediaries are involved until proven otherwise.

- If OPES intermediaries are suspected, request OPES traces from potentially affected user(s). The trace will be a part of the application message received by the user software. If users cooperate, the provider(s) have all the information they need. If users do not cooperate, the provider(s) cannot do much about it (they might be able to deny service to uncooperative users in some cases).

- Some traces may indicate that more information is available by accessing certain resources on the specified OPES intermediary or elsewhere. Content providers may query for more information in that case.

- If everything else fails, providers can enforce no-adaptation policy using appropriate OPES bypass mechanisms and/or end-to-end mechanisms.

9. Security considerations

TBD

10. IANA Considerations

The proposed work will evaluate current protocols for OCP. If the work determines that a new protocol need to be developed, then there may be a need to request new numbers from IANA.

Normative References

[1] McHenry, S., et. al, "OPES Scenarios and Use Cases", Internet-Draft TBD, May 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] OPES working group, "OPES Service Authorization and Enforcement Requirements", Internet-Draft TBD, May 2002.

[5] OPES working group, "OPES Ruleset Schema", Internet-Draft TBD, May 2002.

[6] A. Beck et al., "Requirements for OPES Callout Protocols", Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-opes-protocol-reqs-03.txt, December 2002.

[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-00.txt, October 2002.

[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.

Informative References

[9] 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.

[10] 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.

[11] "Hit Metering", RFC .

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

TBD

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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