Network Working Group
Request for Comments: 3838
Category: Informational

A. Barbir
Nortel Networks
O. Batuner
Consultant
A. Beck
Lucent Technologies
T. Chan
Nokia
H. Orman
Purple Streak Development

August 2004

Policy, Authorization, and Enforcement Requirements of the Open Pluggable Edge Services (OPES)

Status of this Memo

Copyright Notice

Abstract

This document describes policy, authorization, and enforcement requirements for the selection of the services to be applied to a given Open Pluggable Edge Services (OPES) flow.

Table of Contents

1. Introduction

The Open Pluggable Edge Services (OPES) [1] architecture 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 OPES processor can distribute the responsibility of service execution by communicating and collaborating with one or more remote callout servers.

The execution of such services is governed by a set of rules installed on the OPES processor. The rule evaluation can trigger the execution of service applications local to the OPES processor or on a remote callout server.

Policies express the goals of an OPES processor as a set of rules used to administer, manage, and control access to resources. The requirements in this document govern the behavior of OPES entities in determining which of the available services are to be applied to a given message, if any.

The scope of OPES policies described in this document are limited to those that describe which services to call and, if appropriate, with what parameters. These policies do not include those that prescribe the behavior of the called services. It is desirable to enable a common management framework for specifying policies for both the calling of and the behavior of a service. The integration of such a function is the domain of policy administration user interaction applications.

The document is organized as follows: Section 2 considers policy framework. Section 3 discusses requirements for interfaces, while section 4 examines authentication of principals and authorization of services.

2. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [4]. When used with the normative meanings, these keywords will be all uppercase. Occurrences of these words in lowercase comprise normal prose usage, with no normative implications.

3. Policy Architecture

This section describes the architectural policy decomposition requirements. It also describes the requirements for the interfaces between the policy components. Many of the rules here were determined under the influence of RFC 3238 [2].

3.1. Policy Components and Functions

The policy functions are decomposed into three components: a Rule Author, a Policy Decision Point (PDP) [6], and a Policy Enforcement Point (PEP) [6]. The Rule Author provides the rules to be used by an OPES entity. These rules control the invocation of services on behalf of the rule author. The PDP and the PEP interpret the collected rules and appropriately enforce them. The decomposition is illustrated in Figure 1.

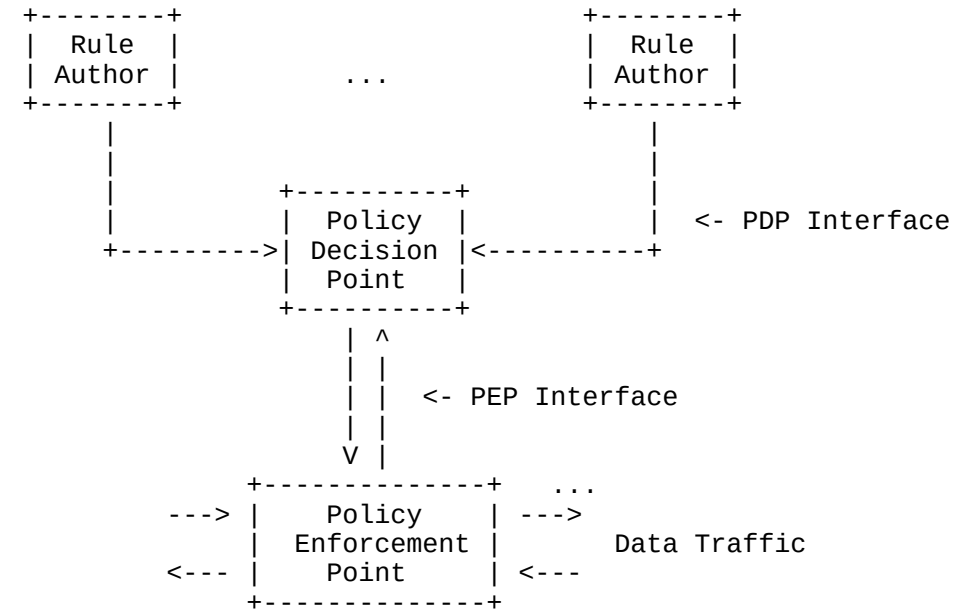

Figure 1: Policy Components

The decomposition of policy control into a PDP and a PEP permit the offloading of some tasks to an administrative service that may be located on a server separate from the real-time enforcement services of the PEP that reside on the OPES processor.

The PDP provides for the authentication and authorization of rule authors and the validation and compilation of rules.

The PEP resides in the data filter where the data from an OPES flow is evaluated against the compiled rules and appropriate calls to the requested services are performed.

Interfaces between these architectural components are points of interoperability. The interface between rule authors and the policy decision points (PDP Interface) MUST use the format that may result from the requirements as described in this document.

The interface between the policy decision points and the policy enforcement points (PEP Interface) can be internal to a specific vendor implementation of an OPES processor. Implementations MUST use standard interface only if the PDP and the PEP reside on different OPES processors.

3.2. Requirements for Policy Decision Points

The Policy Decision Point is essentially a policy compiler. The PDP MUST be a service that provides administrative support to the enforcement points. The PDP service MUST authenticate the rule authors.

The PDP MUST verify that the specified rules are within the scope of the rule authors authority. The PDP MUST be a component of the OPES Administration Authority.

3.3. Requirements for Policy Enforcement Points

In the OPES architecture, the data filter represents a Policy Enforcement point (PEP). At this point, data from an OPES flow is evaluated against the compiled rules, and appropriate calls to the requested services are performed.

In the PEP rules MAY chain actions together, where a series of services to be called are specified. Implementation MUST ensure the passing of information from one called service to another. Implementation MUST NOT prohibit the re-evaluation of a message to determine if another service or set of services should be called.

The execution of an action (i.e., the triggering of a rule) may lead to the modification of message property values. For example, an OPES service that under some circumstances converts JPEG images to GIF images modifies the content type of the requested web object.

Such modification of message property values may change the behavior of subsequently performed OPES actions. The data filter SHOULD act on matched rules before it evaluates subsequent rules. Multiple matched rules can be triggered simultaneously if the data filter can determine in advance that there are no side effects from the execution of any specific rule.

A data filter MAY evaluate messages several times in the course of handling an OPES flow. The rule processing points MAY be defined by administratively defined names. The definition of such names can serve as a selector for policy rules to determine the applicability of a rule or a set of rules at each processing point.

Policy roles ([5] and [6]) SHOULD be used where they aid in the development of the OPES policy model.

Figure 2 expresses a typical message data flow between a data consumer application, an OPES processor, and a data provider application. There are four commonly used processing points identified by the numbers 1 through 4.

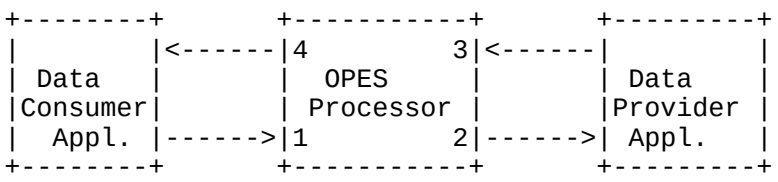

Figure 2: Processing Execution Points

Any data filter (PEP) or any administrative (PDP) implementation MUST support the four rule processing points.

- Data Consumer Request handling role: This involves request processing when received from a Data Consumer Application.
- OPES Processor Request handling role: This involves request processing before forwarding to Data Provider Application.
- Data Provider Response handling role: This involves response processing when forwarding to Data Consumer Application.
- OPES Processor Response handling role: This involves response processing when forwarding to Data Consumer Application.

4. Requirements for Interfaces

The interface between the policy system and OPES services needs to include the ability to pass system state information as well as the subject message.

4.1. Service Bindings Requirements

The invoked OPES services MUST be able to be specified in a location independent fashion. That is, the rule authors need not know and need not specify the instance of an OPES service in the rules.

The rule author SHOULD be able to identify the required service at the detail level that is appropriate for his or her needs. The rule author SHOULD be able to specify a type of service or be able to specify any service that fits a general category of service to be applied to its traffic.

The binding of OPES service names to a specific service MAY be distributed between the PDP and the PEP. As rules are compiled and validated by the PDP, they MUST be resolved to a specific installations' set of homogeneous OPES service.

The selection of a specific instance MAY be postponed and left to PEP to select at either the rule installation time or at run time. To achieve interoperability, PEP MUST support resolving a generic name to a specific instance. It is possible to use services such as SLP or UDDI to resolve generic service names to specific OPES service instances.

The policy system MAY support dynamic discovery of service bindings. The rule author may not know specific service bindings, such as protocol and parameters, when a rule (as specified on the PDP Interface) is general in nature. The required binding information MUST be provided by the PDP and conveyed on the PEP Interface. A service description methodology such as WSDL [8] MUST be present in the policy system.

4.1.1. Environment Variables

There may be a need to define and support a means for maintaining state information that can be used in both condition evaluation and action execution. Depending on the execution environment, OPES services MAY have the freedom to define variables that are needed and use these variables to further define their service behavior without the data filter support.

4.1.2. Requirements for Using State Information

Policy rules MAY specify that state information be used as part of the evaluation of the rules against a given message in an OPES flow. Thus, the policy system SHOULD support the maintenance of groups that can be used in evaluating rule conditions. Membership in such groups can be used as action triggers.

For example, an authorized site blocking service might conclude that a particular user shouldn't be permitted access to a certain web site. Rather than calling the service for each request sent by such a user, a rule might be created to determine whether a user is a member of blocked users and if a requested site is a member of blocked-sites, and then invoke a local blocking service to return an appropriate message to the user.

4.1.3. Requirements for Passing Information Between Services

Environment variables can be used to pass state information between services. For example, analysis of the request or modifications to the request may need to be captured as state information that can be passed to other services on the request path or to services on the response(s) associated with that request.

In the PEP, there SHOULD be provisions to enable setting up variables when returning from a service call and passing variables to other called services based on policy.

4.2. Requirements for Rule and Rules Management

This section provides the requirements for rule management. The rules are divided into two groups. Some rules are provided by the data consumer application, and other rules are provided by the data provider application.

4.2.1. Requirements for Rule Providers

The requirements for rule providers are:

- Rule providers MUST be authenticated and authorized for rules that apply to their network role.
- Rule providers MUST NOT be able to specify rules that are NOT within their scope of authority.
- Rule providers SHOULD be able to specify only what is needed for their services.
- Compilation of rules from different sources MUST NOT lead to execution of conflicting rules.
- The resolution of such rule conflicts is out of scope.

- o Rules are assumed to be static and applied to current network state.

4.2.2. Requirements for Rule Formats and Protocols

It is desirable to choose standard technologies like XML to specify the rule language format.

Rules need to be sent from the rule authors to the OPES administrative server for service authorization, rule validation, and compilation. The mechanisms for doing that are out of scope of the current work.

Once the rules are authorized, validated, and compiled by the administrative server, the rules need to be sent to the OPES processor. The mechanisms for doing that are out of scope of the current work.

4.2.3. Requirements for Rule Conditions

Rule conditions MUST be matched against attribute values of the encapsulated protocol as well as environment variable values. Attribute values of the encapsulated protocol include protocol header values and possibly also protocol body values.

Some OPES services may need to be invoked for all user requests or server responses, such as services with logging functionality, for example. The rule system SHOULD allow unconditional rules rather than requiring rule authors to specify rule conditions that are always true.

4.2.4. Requirements for Rule Actions

The rule system MUST allow for the specification of rule actions that are triggered if the conditions of a rule are met. Matched rules typically lead to the invocation of local or remote services. Rule actions MUST identify the OPES service that is to be executed for the current message request or response.

Rule actions MAY contain run-time parameters which can be used to control the behavior of an OPES service. If specified, these parameters MUST be passed to the executed OPES service.

4.3. Requirements for Policy Expression

OPES processors MUST enforce policy requirements set by data consumers and/or data publishers in accordance with the architecture [1] and this document. They cannot do this consistently unless there are an unambiguous semantics and representation of the data elements mentioned in the policy. For example, this document mentions protection of user "identity" and "profile" information. If a user specifies that his identity must not be shared with other OPES administrative trust domains, and later discovers that his family name has been shared, he might complain. If he were told that "family names are not considered 'identities' by this site", he would probably feel that he had cause for complaint. Or, he might be told that when he selected "do not share identity" on a web form offered by the OPES service provider, that this only covered his login name, and that a different part of the form had to be filled out to protect the family name. A further breakdown can occur if the configuration information provided by such a web form gets translated into configuration elements given to an OPES processor, and those configuration elements are difficult for a software engineer to translate into policy enforcement. The data elements might have confusing names or be split into groupings that are difficult to relate to one another.

The examples illustrate why the OPES policy MUST have definitions of data elements, their relationships, and how they relate to enforcement. These semantics of essential items do not require a separate protocol, but they MUST be agreed upon by all OPES service providers, and the users of OPES services MUST be assured that they have the ability to know their settings, to change them if the service provider policy allows the changes, and to have reasonable assurance that they are enforced with reasonable interpretations.

The requirements for policy data elements in the OPES specification do not have to be all-inclusive, but they MUST cover the minimal set of elements that enable the policies that protect the data of end users and publishers.

5. Authentication of Principals and Authorization of Services

This section considers the authorization and authentication of OPES services.

5.1. End Users, Publishers and Other Considerations

5.1.1. Considerations for End Users

An OPES rule determines which attributes of traffic will trigger the application of OPES services. The author of the service can supply rules, but the author cannot supply the necessary part of the rule precondition that determines which network users will have the OPES services applied for them. This section discusses how users are identified in the rule preconditions, and how users can select and deselect OPES services for their traffic, how an OPES service provider SHOULD identify the users, and how they determine whether or not to add their service selection to an OPES enforcement point.

An OPES service provider MUST satisfy these major requirements:

- Allow all users to request addition, deletion, or blocking of OPES services for their traffic (blocking means "do not use this service for my traffic").
- Prevent untrusted users from causing OPES services to interfere with the traffic of other users.
- Allow users to see their OPES service profiles and notify them of changes.
- Keep a log of all profile activity for audit purposes.
- Adhere to a privacy policy guarding users' profiles.

The administrator of the PDP is a trusted party and can set policy for individuals or groups using out-of-band communication and configuration files. However, users MUST always be able to query the PDP in order to learn what rules apply to their traffic.

Rules can be deposited in the PDP with no precondition relating to network users. This is the way rules are packaged with an OPES service when it is delivered for installation. The PDP is responsible for binding identities to the rules and transmitting them to the PEP. The identity used by the PDP for policy decisions MUST be strictly mapped to the identity used by the PEP. Thus, if a user goes through an identification and authentication procedure with the PDP and is known by identity "A", and if the PEP uses IP addresses for identities, then the PDP MUST provide the PEP with a binding between "A" and A's current IP address.

5.1.2. Considerations for Publishing Sites

An OPES service provider acting on behalf of different publishing sites SHOULD keep all the above considerations in mind when implementing an OPES site. Because each publishing site may be represented by only a single identity, the authentication and authorization databases may be easier for the PEP to handle.

5.1.3. Other Considerations

Authentication may be necessary between PDP's and PEP's, PEP's and callout servers, PEP's and other PEP's, and callout servers and other callout servers, for purposes of validating privacy policies. In any case where user data or traffic crosses trust domain boundaries, the originating trust domain SHOULD have a policy describing which other domains are trusted, and it SHOULD authenticate the domains and their policies before forwarding information.

5.2. Authentication

When an individual selects (or deselects) an OPES service, the individual MUST be authenticated by the OPES service provider. This means that a binding between the user's communication channel and an identity known to the service provider is made in a secure manner. This SHOULD be done using a strong authentication method with a public key certificate for the user; this will be helpful in resolving later disputes. It is recommended that the service provider keep a log of all requests for OPES services. The service provider SHOULD use public key certificates to authenticate responses to requests.

The service provider may have trusted users who through explicit or implicit contract can assign, remove, or block OPES services for particular users. The trusted users MUST be authenticated before being allowed to take actions which will modify the policy base, and thus, the actions of the PEP's.

Because of the sensitivity of user profiles, the PEP Interface between the PEP and the PDP MUST use a secure transport protocol. The PEP's MUST adhere to the privacy preferences of the users.

When an OPES service provider accepts an OPES service, there MUST be a unique name for the service provided by the entity publishing the service. Users MAY refer to the unique name when requesting a service. The unique name MUST be used when notifying users about their service profiles. PEP's MUST be aware of the unique name for each service that can be accessed from their domain. There MUST be a cryptographic binding between the unique name and the entity

responsible for the functional behavior of the service, i.e., if it is a human language translating service, then the name of company that wrote the software SHOULD be bound to the unique name.

5.3. Authorization

In addition to requesting or terminating specific services, users MAY block particular services, indicating that the services should not be applied to their traffic. The "block all OPES" directive MUST be supported on a per user basis.

A response to a request for an OPES service can be positive or negative. Reasons for a negative response include "service unknown" or "service denied by PDP policy". Positive responses SHOULD include the identity of the requestor and the service and the type of request.

As described in the OPES Architecture [1], requests for OPES services originate in either the end user or the publisher domain. The PDP bases its authorization decision on the requestor and the domain. There are some cases where the decision may be complicated.

- The end user has blocked a service, but a trusted user of the PDP wants it applied anyway. In this case, the end user SHOULD prevail, unless there are security or legal reasons to leave it in place.
- The publisher and the end user are in the same domain. If the publisher and end user are both clients of a PDP, can they make requests that effect each other's processing? In this case, the PDP MUST have policy rules naming the identities that are allowed to set such rules.
- The publisher requests a service for an end user. In this case, where the PDP and PEP are in the publisher's administrative domain, the publisher has some way of identifying the end user and his traffic, and the PDP MUST enable the PEP to enforce the policy. This is allowed, but the PDP MUST use strong methods to identify the user and his traffic. The user MUST be able to request and receive information about the service profile that a publisher site keeps about him.
- The end user requests a service specific to a publisher's identity (e.g., nfl.com), but the publisher prohibits the service (e.g., through a "NO OPES" application header). As in the case above, the publisher MUST be able to request and receive profile information that a user keeps about a publisher.

In general, the PDP SHOULD keep its policy base in a manner that makes the decision procedure for all cases easy to understand.

5.4. Integrity and Encryption

5.4.1. Integrity and Confidentiality of Authentication and Requests/ Responses for Service

The requests and responses SHOULD be cryptographically tied to the identities of the requestor and responder, and the messages SHOULD NOT be alterable without detection. A certificate-based digital signature is strongly recommended as part of the authentication process. A binding between the request and response SHOULD be established using a well-founded cryptographic means, to show that the response is made in reply to a specific request.

5.4.2. Integrity and Confidentiality of Application Content

As directed by the PEP, content will be transformed in whole or in part by OPES services. This means that end-to-end cryptographic protections cannot be used. This is probably acceptable for the vast majority of traffic, but in cases where a lesser form of content protection is desirable, hop-by-hop protections can be used instead. The requirements for such protections are:

- Integrity using shared secrets MUST be used between all processing points, end-to-end (i.e., the two ends of a "hop" MUST share a secret, but the secret can be different between "hops"). The processing points include the callout servers.
- Encryption can be requested separately, with the same secret sharing requirement between "hops". When requested, encryption applies to all processing points, including callout servers.
- The signal for integrity (and optionally encryption) MUST originate from either the requestor (in which case it is applied to the response as well) or the responder (in which case it covers only the response).
- The shared secrets MUST be unique (to within a very large probabilistic certainty) for each requestor/responder pair. This helps to protect the privacy of end user data from insider attacks or configuration errors while it transits the provider’s network.

5.5. Privacy

The PDP MUST have a privacy policy regarding OPES data such as user profiles for services. Users MUST be able to limit the promulgation of their profile data and their identities.

Supported limitations MUST include:

- The ability to prevent Identity from being given to callout servers.

- The ability to prevent Profile information from being shared.
- The ability to prevent Traffic data from being sent to callout servers run by third parties.
- The ability to prevent Traffic from particular sites from being given to OPES callout servers.

When an OPES service is provided by a third-party, it MUST have a privacy policy and identify itself to upstream and downstream parties, telling them how to access its privacy policy. A mechanism is needed to specify these preferences and a protocol to distribute them (see section 3.3).

6. Security Considerations

This document discusses policy, authorization and enforcement requirements of OPES. In [3] multiple security and privacy issues related to the OPES services are discussed.

7. References

7.1. Normative References

[1] Barbir, A., Penno, R., Chen, R., Hofmann, M., and H. Orman, "An Architecture for Open Pluggable Edge Services (OPES)", RFC 3835, August 2004.

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

[3] Barbir, A., Batuner, O., Srinivas, B., Hofmann, M., and H. Orman, "Security Threats and Risks for Open Pluggable Edge Services (OPES)", RFC 3837, August 2004.

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

[5] Moore, B., Ellesson, E., Strassner, J., and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.

7.2. Informative References

[6] 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.

[7] 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.

[8] Christensen, et al., Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

8. Acknowledgements

Many thanks to Andreas Terzis, L. Rafalow (IBM), L. Yang (Intel), M. Condry (Intel), Randy Presuhn (Mindspring), and B. Srinivas (Nokia).

9. Authors' Addresses

Abbie Barbir
Nortel Networks
3500 Carling Avenue
Nepean, Ontario K2H 8E9
Canada
Phone: +1 613 763 5229
EMail: abbieb@nortelnetworks.com

Oskar Batuner
Consultant
EMail: batuner@attbi.com

Andre Beck
Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733
USA
EMail: abeck@bell-labs.com

Tat Chan
Nokia
5 Wayside Road
Burlington, MA 01803
USA
EMail: Tat.Chan@nokia.com

Hilarie Orman
Purple Streak Development
EMail: ho@alum.mit.edu

10. Full Copyright Statement

Intellectual Property

Acknowledgement

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