XACML 3.0 Additional Combining Algorithms Profile Version 1.0

Committee Specification Draft 02

09 January 2014

Specification URIs

This version:

http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csd02/xacml-3.0-combalgs-v1.0-csd02.doc (Authoritative)
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csd02/xacml-3.0-combalgs-v1.0-csd02.html
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csd02/xacml-3.0-combalgs-v1.0-csd02.pdf

Previous version:

http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csprd01/xacml-3.0-combalgs-v1.0-csprd01.doc (Authoritative)
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csprd01/xacml-3.0-combalgs-v1.0-csprd01.html
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csprd01/xacml-3.0-combalgs-v1.0-csprd01.pdf

Latest version:

http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/xacml-3.0-combalgs-v1.0.doc (Authoritative)
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/xacml-3.0-combalgs-v1.0.html
http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/xacml-3.0-combalgs-v1.0.pdf

Technical Committee:

OASIS eXtensible Access Control Markup Language (XACML) TC

Chairs:

Hal Lockhart (hal.lockhart@oracle.com), Oracle
Bill Parducci (bill@parducci.net), Individual

Editor:

Erik Rissanen (erik@axiomatics.com), Axiomatics

Related work:

This specification is related to:

- *eXtensible Access Control Markup Language (XACML) Version 3.0.* Edited by Erik Rissanen. 22 January 2013. OASIS Standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

Abstract:

This profile defines new useful but optional combining algorithms for XACML 3.0.

Status:

This document was last revised or approved by the OASIS eXtensible Access Control Markup Language (XACML) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/xacml/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/xacml/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[xacml-3.0-combalgs]

XACML 3.0 Additional Combining Algorithms Profile Version 1.0. Edited by Erik Rissanen. 09 January 2014. OASIS Committee Specification Draft 02. http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/csd02/xacml-3.0-combalgs-v1.0-csd02.html. Latest version: http://docs.oasis-open.org/xacml/xacml-3.0-combalgs/v1.0/xacml-3.0-combalgs-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

This profile defines additional combining algorithms for XACML 3.0, **[XACML3]**. These algorithms may be useful in certain contexts, but have not been considered important enough to include as mandatory items in the core XACML specification.

1.1 Terminology

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 **[RFC2119]**.

1.2 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[XACML3] *eXtensible Access Control Markup Language (XACML) Version 3.0*, 22 January 2013. OASIS Standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.doc

1.3 Non-Normative References

None

2 on-permit-apply-second policy combining algorithm

2.1 Algorithm definition

This section defines the “on-permit-apply-second” policy combining algorithm of a policy set.

The policy combining algorithm defined here has the following identifier:

urn:oasis:names:tc:xacml:3.0:policy-combining-algorithm:on-permit-apply-second

The following is a non-normative informative description of this combining algorithm:

> The on permit deny second combining algorithm is primarily intended for those cases where it would be desirable to attach a condition to a policy or policy set. This algorithm has the following behavior.
>
> The input contains an array of children (policies and/or policy sets).
>
> 1. If there are not exactly two or three children, then the result is "Indeterminate{DP}".
> 2. Otherwise, if the decision from the first child is "NotApplicable", "Deny", or "Indeterminate{D}", then the result is "NotApplicable" if there is no third child, or the decision of the third child if there is a third child.
> 3. Otherwise, if the decision from the first child is "Permit", then the result is the decision from the second child.
> 4. Otherwise, the result is "Indeterminate{DP}".

The following pseudo-code represents the normative specification of this policy combining algorithm. The algorithm is presented here in a form where the input to it is an array with children (the policies or policy sets) of the policy set.

```
Decision onPermitApplySecondCombiningAlgorithm(Node[] children)
{
    if (lengthOf(children) < 2 || lengthOf(children) > 3) {
        // Use status code
        // urn:oasis:names:tc:xacml:1.0:status:processing-error
        return Indeterminate{DP}
    }
    Decision decision0 = children[0].evaluate();
    if (decision0 == NotApplicable ||
        decision0 == Deny ||
        decision0 == Indeterminate{D}) {
        if(lengthOf(children) == 2) {
          return NotApplicable;
        }
        Decision decision2 = children[2].evaluate();
        return decision2;
    }
    if (decision0 == Permit) {
        Decision decision1 = children[1].evaluate();
        return decision1;
    }
    // decision0 is Indeterminate{P} or Indeterminate{DP}
    // Use status code of decision0
    return Indeterminate{DP};
}
```

Obligations and advice MUST be combined as described in **[XACML3]**.

2.2 Discussion (non-normative)

XACML 3.0, **[XACML3]**, does not allow <Condition> elements at the policy or policy set levels. In some cases it may be useful to have a <Condition> at the policy or policy set level since a <Condition> allows for more expressive matching than a <Target>, which can only match against constant values.

For instance, someone may want to write a policy which applies to the cases where the subject is the owner of the resource. In this case the policy should apply if the subject-id of the request equals the owner attribute of the resource in the request. This matching cannot be done with a <Target> since it is not a match expression against a constant value. Such a policy would require a <Condition> at the Policy level.

The on-permit-apply-second combining algorithm makes it possible to define a policy structure which behaves as if there was a <Condition> at the policy or policy set level, without changes to the XACML 3.0 schema.

For instance, assume that someone wants to define policy A, which should contain condition A. Ideally the user would like to define this policy:

```
Policy A:
  Target A
  Condition A
  Rules of A...
```

This is not possible in XACML 3.0, but with the help of the on-permit-apply-second combining algorithm, the above policy structure can be refactored into the following structure, which has the desired effect:

```
PolicySet X [on-permit-apply-second]
  Target A
  Policy Y
    Rule Z [Permit]
      Condition A
  Policy A:
    Rules of A...
```

If Target A matches, then the on-permit-apply-second combining algorithm will evaluate policy Y. If Condition A applies, policy Y will say Permit and policy A is evaluated and the result of policy A is used as the result of policy set X. If Condition A does not apply, then policy set X returns NotApplicable.

A similar structure can be used to get the effect of a <Condition> in a <PolicySet>.

Likewise there is no combining algorithm in XACML 3.0, **[XACML3]**, which ensures that if a specific branch of a policy tree has been evaluated, then no other branch is evaluated, even if the first branch would evaluate to NotApplicable. The on-permit-apply-second algorithm can take a third child which will be used only in the case the second child is not selected by the condition, similarly to an “if-then-else” construct which is available in many computer languages. In the following example, depending on condition C, either policy A or policy B is evaluated, but in no case are both A and B evaluated.

```
PolicySet X [on-permit-apply-second]
  Policy
    Rule [Permit]
      Condition C
  Policy A:
    Rules of A...
  Policy B:
    Rules of B...
```

3 Conformance

The following table lists the defined algorithms in this profile. Each of them is optional to implement so an implementation may choose to implement and conform to one or more of the described combining algorithms depending on the usefulness of the algorithm in the implementation context.

urn:oasis:names:tc:xacml:3.0:policy-combining-algorithm:on-permit-apply-second

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

David Brossard, Axiomatics
Erik Rissanen, Axiomatics
Remon Sinnema, EMC
David Staggs, Jericho Systems
Danny Thorpe, Quest Software
Crystal Hayes, The Boeing Company
Richard Hill, The Boeing Company
John Tolbert, The Boeing Company
Jean-Paul Buu-Sao, Transglobal Secure Collaboration Participation, Inc. (TSCP)
Bill Parducci, Individual Member
Hal Lockhart, Oracle
Rich Levinson, Oracle

Appendix B. Revision History

Revision	Date	Editor	Changes Made
WD 01	17 Feb 2012	Erik Rissanen	Initial version with the on-permit-apply-second algorithm.
WD 02	22 Aug 2012	Erik Rissanen	Added acknowledgements
WD 03	31 Oct 2012	Erik Rissanen	Changed the example for refactoring conditions into a slightly different form. Generalized the definition of the on-permit-apply-second algorithm based on discussion on the XACML TC list.
WD 04	19 Jun 2013	Erik Rissanen	Generalized the on-permit-apply-second algorithm to also accept two children. Simplified handling of Indeterminate. Updated cross reference to XACML 3.0 OASIS standard version.
WD 05	27 Jun 2013	Erik Rissanen	Changed the return value of on-permit-apply-second in case of Indeterminate
WD 06	16 Dec 2013	Erik Rissanen	Corrected the non-normative description of on-permit-apply-second to match the change made in WD05.