NIST Special Publication 800-178

A Comparison of Attribute Based Access Control (ABAC) Standards for Data Service Applications

Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC)

David Ferraiolo
Ramaswamy Chandramouli
Vincent Hu
Rick Kuhn

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-178

C O M P U T E R S E C U R I T Y

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 800-178

A Comparison of Attribute Based Access Control (ABAC) Standards for Data Service Applications

Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC)

David Ferraiolo
Ramaswamy Chandramouli
Vincent Hu
Rick Kuhn
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-178

October 2016

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-178
Natl. Inst. Stand. Technol. Spec. Publ. 800-178, 68 pages (October 2016)
CODEN: NSPUE2

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-178

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: sp800-178@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC) are very different attribute based access control (ABAC) standards with similar goals and objectives. An objective of both is to provide a standardized way for expressing and enforcing vastly diverse access control policies on various types of data services. However, the two standards differ with respect to the manner in which access control policies are specified and implemented. This document describes XACML and NGAC, and then compares them with respect to five criteria. The goal of this publication is to help ABAC users and vendors make informed decisions when addressing future data service policy enforcement requirements.

Keywords

access control; access control mechanism; access control model; access control policy; attribute based access control (ABAC); authorization; Extensible Access Control Markup Language (XACML); Next Generation Access Control (NGAC); privilege

Acknowledgements

The authors, David Ferraiolo, Ramaswamy Chandramouli, Vincent C. Hu, and Rick Kuhn of the National Institute of Standards and Technology (NIST), wish to thank their colleagues who reviewed drafts of this document, including the following: Karen Scarfone (Scarfone Cybersecurity), Wayne Jansen (Bayview Behavioral Consulting), Serban Gavrila (NIST), Indrakshi Ray (Colorado State University), Duminda Wijesekera (George Mason University), and Ram Krishnan (University of Texas at San Antonio).

The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

Note to Readers

For purposes of transparency, one of the authors of this document, David Ferraiolo, is a member of the American National Standards Institute/International Committee for Information Technology (ANSI/INCITS) Next Generation Access Control (NGAC) working group. To mitigate the injection of bias, prior drafts have been circulated for review to multiple subject matter experts, as well as formally to the public at large. In addition, prior to publication, the final draft of this document was reviewed by multiple independent entities in compliance with the requirements of the NIST Editorial Review Board (ERB). Disposition and resolution of comments were considered by the authors at large.

Trademark Information

Executive Summary

Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC) are very different attribute based access control (ABAC) standards with similar goals and objectives. XACML, available since 2003, is an Extensible Markup Language (XML) based language standard designed to express security policies, as well as the access requests and responses needed for querying the policy system and reaching an authorization decision [1]. NGAC is a relations and architecture-based standard designed to express, manage, and enforce access control policies through configuration of its relations.

What are the similarities and differences between these two standards? What are their comparative advantages and disadvantages? These questions are particularly relevant because XACML and NGAC are different approaches to achieving a common access control goal—to allow applications with vastly different access policies to be expressed and enforced using the features of the same underlying mechanism in diverse ways. These are also important questions, given the prevalence of data services in computing. Data services include computational capabilities that allow the consumption, alteration, and management of data resources, and distribution of access rights to data resources. Data services can take on many forms, to include applications such as time and attendance reporting, payroll processing, and health benefits management, but also including system level utilities such as file management.

To answer these questions, this document first describes XACML and NGAC, then compares them with respect to five criteria. The first criterion is the relative degree to which the access control functionality of a data service can be separated from a proprietary operational environment. The other four criteria are derived from ABAC issues or considerations identified by NIST Special Publication (SP) 800-162 [2]: operational efficiency, attribute and policy management, scope and type of policy support, and support for administrative review and resource discovery.

Although NGAC is only now emerging as a national standard, it compares favorably in many respects with XACML and should be considered, along with XACML, by both users and vendors in addressing future data service policy enforcement requirements. Below is a summary of this comparison.

Separation of Access Control Functionality from Proprietary Operating Environments

Both XACML and NGAC achieve separation of access control functionality of data services from proprietary operating environments, but to different degrees. XACML's separation is partial. XACML does not envisage the design of a Policy Enforcement Point (PEP) that is data service agnostic. An XACML deployment consists of one or more data services, each with an operating environment-dependent PEP and operating environment-dependent operational routines and resource types that share a common Policy Decision Point (PDP) and access control information consisting of policies and attributes.

The degree of separation that can be achieved by NGAC is near complete. Although an NGAC deployment could include a PEP with an application programming interface (API) that recognizes operating environment-specific operations (e.g., send and forward operations for a

messaging system), it does not necessarily need to do so. NGAC includes a standard PEP with an API that supports a set of generic, operating environment-agnostic operations (read, write, create, and delete policy elements and relations). This API enables a common, centralized PEP to be implemented to serve the requests of multiple applications.

Operational Efficiency

An XACML request is a collection of attribute name, value pairs for the subject (user), action (operation), resource, and environment. XACML identifies relevant policies and rules for computing decisions through a search for Targets (conditions that match the attributes of the request). Because multiple Policies in a PolicySet and/or multiple Rules in a Policy may produce conflicting access control decisions, XACML resolves these differences by applying a policy combining algorithm from a set defined by the standard. The entire process includes collecting attributes, matching conditions, computing rules, and resolving conflicts involving at least two data stores. There are two phases of policy evaluation that need to be considered. The first and costliest is loading policy from disk to Policy Decision Point (PDP) main memory, and the second is request evaluation. In both phases, performance is directly related to the number of policies considered.

An NGAC request is composed of a process id, user id, operation, and a sequence of one or more operands mandated by the operation that affects either a resource or access control data. NGAC identifies relevant Policies and attributes by reference when computing a decision. NGAC computes decisions by applying a single combining algorithm over applicable Policies that do not conflict. Unlike XACML, NGAC does not need to load policy from disk into memory when evaluating a request. Instead, and as treated in NGAC reference implementation version 1.6 [11] all information necessary in computing an access decision can reside in memory. Memory is initially loaded when the PDP is initialized, and is updated when an administrative change occurs. The NGAC specification describes what constitutes a valid implementation, but does not provide implementation guidance, thereby leaving room for multiple competing approaches with different efficiencies. A measure of the operational efficiency is the complexity of algorithm used for arriving at a policy decision. In its reference implementation Version 1. 6 on GitHub [11], the NGAC computes a decision through an algorithm [30] that is linear. Furthermore, it is not linear in relation to the entire access control data set, but only to the portion relevant to a particular user.

Attribute and Policy Management

Proper enforcement of data resource policies is dependent on administrative policies. This is especially true in a federated or collaborative environment, where governance policies require different organizational entities to have different responsibilities for administering different aspects of policies and their dependent attributes.

XACML and NGAC differ dramatically in their ability to impose policy over the creation and modification of access control data (attributes and policies). NGAC manages attributes and policies through a standard set of administrative operations, applying the same enforcement interface and decision making function as it uses for accessing data resources. XACML does not recognize administrative operations, but instead manages policy content through a Policy

Administration Point (PAP) with an interface that is different from that for accessing data resources. XACML provides support for decentralized administration of some of its access policies. However, the approach is only a partial solution in that it is dependent on trusted and untrusted policies, where trusted policies are assumed valid, and their origin is established outside the delegation model. Furthermore, the XACML delegation model does not provide a means for imposing policy over modification of access policies, and offers no direct administrative method for imposing policy over the management of its attributes.

NGAC enables a systematic and policy-preserving approach to the creation of administrative roles and delegation of administrative capabilities, beginning with a single administrator and an empty set of access control data, and ending with users with data service, policy, and attribute management capabilities. NGAC provides users with administrative capabilities down to the granularity of a single configuration element, and it can deny users administrative capabilities down to the same granularity.

Scope and Type of Policy Support

Although resources may be protected under a wide variety of different access policies, these policies can be generally categorized as either discretionary or mandatory controls. Discretionary access control (DAC) is an administrative policy that permits system users to allow or disallow other users' access to resources that are placed under their control. Although XACML can theoretically provide users with administrative capabilities necessary to control and give away access rights to other users, the approach is complicated by the need to create and maintain additional metadata for each and every object/resource (e.g., Owner attribute). Conversely, NGAC has a flexible means of providing users with administrative capabilities to include those necessary for the establishment of DAC policies.

In contrast to DAC, mandatory access control (MAC) enables ordinary users' capabilities to execute operations on resources, but not administrative operations that may influence those capabilities. MAC policies unavoidably impose rules on users in performing operations on resources. MAC policies can be further characterized as controls that accommodate confinement properties to prevent indirect leakage of data to unauthorized users, and those that do not.

Expression of non-confinement MAC policies is perhaps XACML's strongest suit. XACML can specify rules and other conditions in terms of attribute values of varying types. There are undoubtedly certain policies that are expressible in terms of these rules that cannot be easily accommodated by NGAC. This is especially true when treating attribute values as integers. For example, to approve a purchase request may involve adding a person's credit limit to the person's account balance. Furthermore, XACML takes environmental attributes into consideration in expressing policy, and NGAC does not. However, there are some non-confinement MAC properties, including a variety of history-based policies, that NGAC can express but XACML cannot.

In contrast to NGAC, XACML does not recognize the capabilities of a process independent of the capabilities of its user. Without such features, XACML is ill-equipped to support confinement and as such is arguably incapable of enforcement of a wide variety of policies. These confinement-dependent policies include some instances of role-based access control

(RBAC), e.g., “only doctors can read the contents of medical records,” originator control (ORCON) and Privacy, e.g., “I know who can currently read my data or personal information”, or conflict of interest, e.g., “a user with knowledge of information within one dataset cannot read information in another dataset”. Through imposing process level controls in conjunction with event-response relations, NGAC has shown [3] support for these and other confinement-dependent MAC controls.

Administrative Review and Resource Discovery

A desired feature of access controls is review of capabilities of users and access control entries of objects [4] [18]. These features are often referred to as “before the fact audit” and resource discovery. “Before the fact audit” is one of RBAC’s most prominent features [5]. Being able to discover or see a newly accessible resource is an important feature of any access control system. NGAC supports efficient algorithms for both per-user and per-object review. Per-object review of access control entries is not as efficient as a pure access control list (ACL) mechanism, and per-user review of capabilities is not as efficient as that of RBAC. However, this is due to NGAC’s consideration of conducting review in a multi-policy environment. NGAC can efficiently support both per-object and per-user reviews of combined policies [30], where RBAC and ACL mechanisms can do only one type of review efficiently, and logical formula-based mechanisms such as XACML, although able to combine policies, cannot do either type of review efficiently [6].

Table of Contents