



Attribute-Based Access Control

Vincent C. Hu, D. Richard Kuhn, and David F. Ferraiolo,
National Institute of Standards and Technology

Attribute-based access control (ABAC) is a flexible approach that can implement AC policies limited only by the computational language and the richness of the available attributes, making it ideal for many distributed or rapidly changing environments.

Traditionally, access control (AC) has been based on the identity of a user requesting execution of a capability to perform an operation (for example, read) on an object (for example, a file), either directly or through predefined attribute types such as roles or groups assigned to that user. Practitioners have noted that this AC approach is often cumbersome to manage given the need to associate capabilities directly to users or their roles or groups. In addition, the requester qualifiers of identity, groups, and roles are often insufficient in expressing real-world AC policies. An alternative is to grant or deny user requests based on arbitrary attributes of the user and selected attributes of the object, and environment conditions that could be globally recognized and more relevant to the policies at hand. This approach is often referred to as attribute-based access control (ABAC).

into an AC decision and thereby providing a larger set of possible combinations of those variables to reflect a larger and more definitive set of possible rules to express policies, which are limited only by the computational language and the richness of the available attributes.

This flexibility enables creation of access rules without specifying individual relationships between each subject and each object. For example, a subject is assigned a set of subject attributes upon employment, such as Nancy Smith is a Nurse Practitioner in the Cardiology Department. An object is assigned its object attributes upon creation, such as a folder with Medical Records of Heart Patients. Objects may receive their attributes either directly from the creator or as a result of automated scanning tools. The administrator or owner of an object creates an AC rule using attributes of subjects and objects to govern the set of allowable capabilities—for example,

ABAC: A FLEXIBLE ACCESS CONTROL MODEL

ABAC is a logical AC model that controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request. ABAC enables more precise AC by allowing for a higher number of discrete inputs

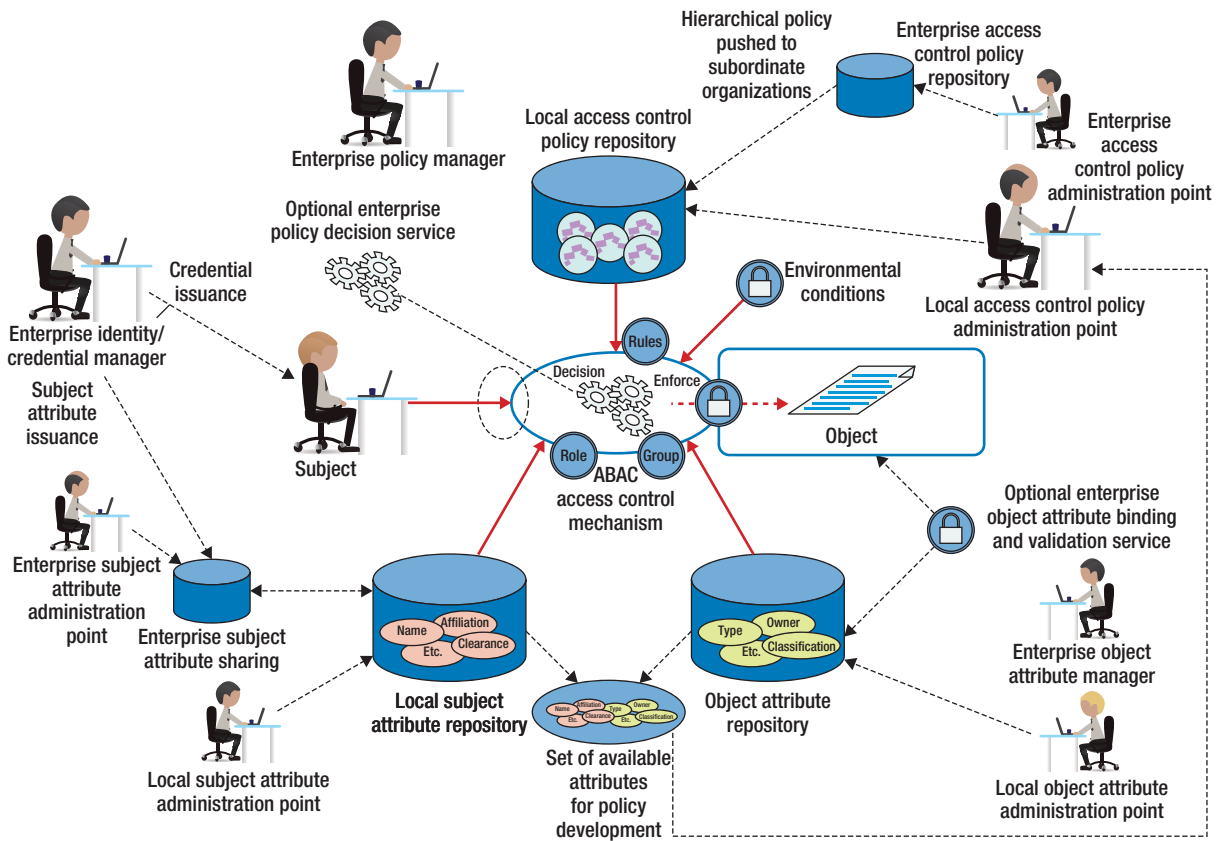


Figure 1. Attribute-based access control (ABAC) example. Adapted from V.C. Hu et al., *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, NIST Special Publication 800-162, Nat'l Institute of Standards and Technology, Jan. 2014.

all Nurse Practitioners in the Cardiology Department can View the Medical Records of Heart Patients.

Under ABAC, access decisions can change between requests simply by altering attribute values, without requiring changes to the subject/object relationships defining the underlying rule sets. This provides a more dynamic AC management capability and limits long-term maintenance requirements of object protections.

Further, ABAC enables object owners or administrators to apply AC policy without prior knowledge of the specific subject and for an unlimited number of subjects that might require access. As new subjects join the organization, rules and objects need not be modified, and as long as the subject is assigned the attributes necessary for access to the required objects—for example, all

Nurse Practitioners in the Cardiology Department are assigned those attributes—no modifications to existing rules or object attributes are required. This accommodation of the external (unanticipated) user is one of the primary benefits of employing ABAC.^{1,2}

As a result of this flexibility, ABAC has attracted interest across industry and government, and is the fastest-growing AC model today.³ It has been integrated with other approaches, such as the International Committee for Information Technology Standards (INCITS) standard for role-based access control,⁴ and has become the basis for an increasing range of products. But beyond the basic scheme of associating attributes with subjects, objects, and environments, there has been little consistency among ABAC implementations.

IMPLEMENTING ABAC IN THE ENTERPRISE ENVIRONMENT

Due to a lack of consensus on ABAC features, users can't accurately assess the benefits and challenges associated with the model. To help address this problem, the National Institute of Standards and Technology (NIST) released Special Publication (SP) 800-162, *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*.¹ This document serves a two-fold purpose. First, it provides federal agencies with a definition of ABAC and a description of its functional components. Second, it describes planning, design, implementation, and operational considerations for employing ABAC within an enterprise to improve information sharing while maintaining control of that information. The guide focuses on the

TABLE 1. Level of attribute assurance (LOAA) mappings example.

LOAA	Accuracy	Integrity	Availability
1	Attributes are properly verified for veracity through provision and management.	Secure attribute repository. Secure communication between attribute providers (APs) and relying parties (RPs).	Attribute refresh frequency meets the system performance requirement.
2	Includes level 1. Documented rule or standards for attribute value assignment and definition (syntax and semantic rule).	Includes level 1. Dedicated attribute repositories.	Includes level 1. Attribute caching during runtime meets the system performance requirement.
3	Includes level 2. Attributes cover all of the organization's protection policy requirements (semantically complete).	Includes level 2. Encrypted attribute values and communications between APs and RPs.	Includes level 2. Failover or backup attributes support.
4	Includes Level 3. Attributes under federated or unified governance.	Includes level 3. Formal rules or policy (or standards) for create, update, modify, and delete attributes.	Includes level 3. Log for attribute changes and access.

challenges of implementing ABAC rather than on balancing the cost and effectiveness of other capabilities versus ABAC.

When deployed across an enterprise to increase information sharing among diverse organizations, ABAC implementations can become complex, requiring an attribute management infrastructure, machine-enforceable policies, and an array of functions that support access decisions and policy enforcement. As Figure 1 shows, in addition to the basic policy, attribute, and AC mechanism requirements, the enterprise must support management functions for enterprise policy development and distribution, enterprise identity and subject attributes, subject attribute sharing, enterprise object attributes, authentication, and AC mechanism deployment and distribution.

Enabling these capabilities requires careful consideration of numerous factors that will influence the design, security, and interoperability of an enterprise ABAC solution. These

factors can be summarized around a set of activities:

- › establish the business case for ABAC implementation;
- › understand the operational requirements and overall ABAC enterprise architecture;
- › establish or refine business processes to support ABAC;
- › develop and acquire an interoperable set of ABAC capabilities; and
- › operate with efficient ABAC processing.

NIST SP 800-162 helps ABAC system planners, architects, managers, and implementers carry out these activities in four phases. The *initiation phase* includes building the business case for deploying ABAC capabilities; scalability, feasibility, and performance requirements; and developing operational requirements and architecture. The *acquisition/development phase* includes business process generation and deployment preparation,

system development and solution acquisition considerations, and other enterprise ABAC capabilities. The *implementation/assessment phase* includes attribute caching, attribute source minimization, and ABAC interface specifications. Finally, the *operations/maintenance phase* includes availability of quality ABAC data.

ATTRIBUTE ASSURANCE

The metadata of ABAC attributes communicate aspects that are important for attribute standardization. By coupling a common set of mandatory and optional metadata with attribute assertions, ABAC systems can query attribute information to make their own risk-based decisions, especially when delivered via a broker connected to many systems.

In general, attribute metadata fall into three categories:

- › *Accuracy* establishes the policy and technical underpinnings for semantically and syntactically correct use of these attributes


and environmental conditions, and ensures that the reported attributes are trustworthy, based on the trust established in the measurement and reporting processes.

- › **Integrity** considers different standards and protocols used for secure sharing of attributes between systems in order to avoid compromising the integrity and confidentiality of the attributes or exposing vulnerabilities in attribute provider (AP) or relying party (RP) systems or entities.
- › **Availability** ensures that the update and retrieval of attributes support the RP. In addition, attribute repositories' failover and backup capability must be considered. Note that some attributes might change regularly or over time.

An AP is any person or system that provides subject, object (or resource), or environmental condition attributes regardless of transmission method. The AP could be the original authoritative source or receiving information from an authoritative source for re-packing and storing-and-forwarding to the ABAC system. Attribute values can be human generated (for example, an employee database) or derived from formulas (for example, a credit score). Regardless of the attribute source, the system should ensure that the attribute value received from an AP is accurately associated with the subject,

object, or environmental condition to which it applies.² Table 1 illustrates example levels of attribute assurance (LOAA) based on the accuracy, integrity, and availability properties.

A ttribute-based access control is a flexible approach that can implement AC policies limited only by the computational language and the richness of the available attributes. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without specifying individual relationships between each subject and each object, making ABAC ideal for many distributed or rapidly changing environments.

ABAC has the potential to dramatically improve AC in modern applications such as e-commerce and the Internet of Things. In the meantime, a consensus definition of ABAC is needed, and work remains to be done in assuring attribute accuracy and reliability. For more information on ongoing efforts, see <http://csrc.nist.gov/projects/abac/index.html>. 

REFERENCES

1. V.C. Hu et al., *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, NIST Special Publication 800-162, Nat'l Institute of Standards and Technology, Jan. 2014; <http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf>.
2. V.C. Hu, D.F. Ferraiolo, and D.R. Kuhn, *Assessment of Access Control Systems*, NIST Interagency Report 7316, Nat'l Institute of Standards and Technology, Mar. 2006; <http://csrc.nist.gov/publications/nistir/7316/NISTIR-7316.pdf>.
3. Avatier Corp., "Leveraging Today's Megatrends to Drive the Future of Identity Management," video presentation, Gartner Identity and Access Management (IAM) Summit, 2012; www.avatier.com/products/identity-management/resources/gartner-iam-2020-predictions.
4. D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role Based Access Control," *Computer*, vol. 43, no. 6, 2010, pp. 79–81.

VINCENT C. HU is a computer scientist in the Computer Security Division at the National Institute of Standards and Technology. Contact him at vhu@nist.gov.

D. RICHARD KUHN is a project leader and computer scientist in the Computer Security Division at the National Institute of Standards and Technology. Contact him at kuhn@nist.gov.

DAVID F. FERRAILO is a computer scientist and manages the Secure Systems and Applications Group in the Computer Security Division at the National Institute of Standards and Technology. Contact him at dferraiolo@nist.gov.



Engineering and Applying the Internet

IEEE Internet Computing

IEEE Internet Computing reports emerging tools, technologies, and applications implemented through the Internet to support a worldwide computing environment.

For submission information and author guidelines, please visit www.computer.org/internet/author.htm