International Journal of Computer Applications Technology and Research
Volume 3– Issue 4, 200 - 208, 2014, ISSN: 2319–8656

AUTHENTICATE SYSTEM OBJECTS USING ACCESS CONTROL POLICY BASED MANAGEMENT

Jeena S Watson
Department of Computer Science and Engineering,
PSN Engineering College,
Tirunelveli, India

R. Natchadalingam
Department of Computer Science and Engineering,
PSN Engineering College,
Tirunelveli, India

Abstract: The network level access control policy is based on policy rule. The policy rule is a basic building of a policy based system. Each policy contains set of conditions and actions. Here conditions are evaluated to determine whether the actions are performed. The existing work is based on packet filtering scenario. Here every policy can be translated into canonical form. That uses the "First Matching Rule" resolution strategy. The access control matrix is proposed to translate the policy. The Generalized Aryabhata Reminder Theorem (GART) is used for to construct the access control matrix. In this access control matrix rows represent users and columns represent files. In which each user is associated with key and each digital file is associated with lock.

Keywords: Policy Algebra Framework; Ubiquitous Enforcement Mechanism; Policy Administration; Policy Matching; Algebraic Framework; Policy Composition and Delegation.

1. INTRODUCTION

The term 'network' is frequently used to describe clusters of different kinds of actor who are linked together in political, social or economic life. Networks may be loosely structured but still capable of spreading information or engaging in collective action. Security in computer systems is based on protecting resources from unauthorized access before that we have to ensure that whether all given requests can be satisfied all the time. The growth of computer systems, both in scale and complexity, so management of the system is very difficult. These systems are often interconnected and form a distributed environment with a large number of devices and users, vast amounts of data and resources, and a variety of applications, protocols, and mechanisms. Policy-based systems management is a very useful for this scenario.

Access Control is any mechanism by which a system grants or revokes the right to access some data, or perform some action. Normally, a user must first login to a system, using some authentication system. Next, the Access Control mechanism controls what operations the user may or may not make by comparing the User ID to an Access Control database. Access Control systems include:

- File permissions, such as create, read, edit or delete on a file server.
- Program permissions, such as the right to execute a program on an application server.
- Data rights, such as the right to retrieve or update information in a database.

A General policy definition adopted in [1] considers policies as rules governing the behavior choices of a system. The policy-driven approach facilitates the dynamic change of behavior of the distributed management system, while avoiding the burden of recoding system functionality upon changes.

1.1 Problem Description

A very important aspect for any policy-based systems management is to protect managed data and resources against unauthorized access, while ensuring their availability to legitimate users. This process is called access control [2]. Access control is a crucial aspect of a system's security, and provides the basis for all the other mechanisms and procedures the system may utilize.

The development of any access control system requires the following two concepts: an access control policy that defines high-level rules according to which access control must be regulated, and an enforcement mechanism that implements the controls imposed by the policy using software and/or hardware solutions.

Given the large number of system elements managed in a distributed environment, the access control mechanism employed must be scalable. The traditional way of dealing with scalability at the human level has been decentralization of management and delegation of authority. Thus it is impossible to maintain a central policy agent for managing all the system devices, which implies the need for integrating and analyzing policies issued by multiple policy authors to ensure that they are always consistent and compliant with the global security requirements.

The goal of policy refinement is to generate low-level rules such that their syntax and semantics can be interpreted by the chosen enforcement mechanism. Given a large number of system elements managed in a distributed scenario, it is efficient and scalable to issue global service and security requirements in terms of high-level policies rather than mechanism rules. On the other hand, these high-level requirements are mostly specified by policy makers without an intimate knowledge of the underlying system.

1.2 Network Access Control

Network access control is concerned with regulating access to protected resource in a communications network that complies with defined security policies. Generally network access control deals with two levels of protection [3]:

- *Host-based security* protects the safety of a single host that is connected to a network. Where hosts within the same administrative zone tend to trust each other such that one weak link can compromise the whole cluster of systems.
- *Perimeter security* protects a cluster of hosts using two components: a layer of defense built up around the

cluster, called the wall, and the gate that allows legitimate traffic to pass through while blocking malicious one. This approach often assumes every host behind the wall is trusted.

Generally a network access control solution unifies a number of mechanisms [4] including, but not limited to, the following techniques

- Endpoint security techniques such as antivirus software to prevent, detect and remove malware such as computer virus, worms, Torjan horses, etc.; host-based intrusion detection and prevention systems that monitor system activities to report malicious behavior and policy violation.
- User or system authentication methods such as passwords (something you know), secure devices (something you own), and biometric (something you have)
- Network security enforcement such as firewalls to protect local system from network-based threat through traffic filtering, IPsec protocols to provide end-to-end or end-to gateway encryption and authentication, etc.

Moreover, there is a growing trend of enforcing access control based on the end-to-end design principle in distributed systems, similar to the IP structure in the communications network. This approach implements access control in a distributed manner by removing potential performance bottleneck to corporate rapidly growing networks, and hence yields better scalability.

1.3 Hypothesis

A policy mechanism containing the following essential elements will provide a more flexible, more efficient, and more secure access control solution [5] for distributed systems,

1. A *distributed policy refinement scheme* automating the translation from high-level security and service requirements into low-level implementable rules as inputs to the enforcement mechanism;

2. A *policy algebra framework* providing a formalism for policy delegation, composition, and analysis in distributed networks, and defining mechanisms to reason about policy languages;

3. A *ubiquitous enforcement mechanism* implementing policy delegation, whose correctness and consistency can be verified using the policy algebra.

The process of authorization is guided by access control policies, and these two terms are often used interchangeably in the context of security policy management. While authorization is concerned with specifying permissions and prohibitions, obligations (or refrain policies) specify management actions that must or must not be performed.

2 POLICY BASED MANAGEMENT

Figure 1.1 depicts the logic flow of policies in the policy based management system. A policy is the combination of rules and services where rules define the criteria for resource access and usage.

Block Diagram:

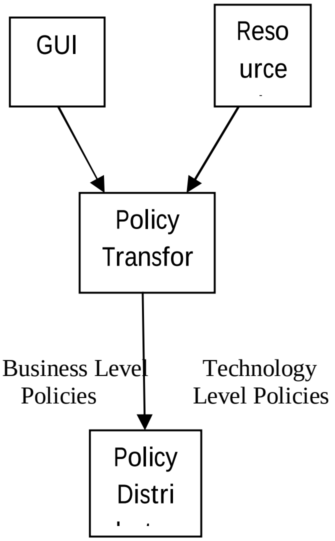

Figure 1.1 A generic policy management tool

A policy is formally defined as an aggregation of policy rules. Each policy rule is composed of a set of conditions and a corresponding set of actions. The condition defines when the policy rule is applicable. Once a policy rule is activated, one or more actions contained by that policy rule may be executed. These actions are associated with either meeting or not meeting the set of conditions specified in the policy rule.

Policy-based systems have become a promising solution for implementing many forms of large-scale, adaptive systems that dynamically change their behavior in response to changes in the environment or to changing application requirements. This can be achieved by modifying the policy rules interpreted by distributed entities, without recoding or stopping the system. Such dynamic adaptability is fundamentally important in the management of increasingly complex computing systems.

Policy-based management (PBM) is a management paradigm that separates the rules governing the behavior of a system from its functionality. It promises to reduce maintenance costs of information and communication systems while improving flexibility and runtime adaptability.

The policy-based technology could relieve the suffering of managing the large computer systems and free the manager from monitoring the equipments and systems directly and supply a systematic method for establishing, revising, and distributing policies. Policy is a kind of criterion that aims at determining the choice of the actions in an individual system. The criterion is long-lasting, illustrative, and originated from the target of the management.

2.1 Policy Specification

To configure access control mechanisms, a number of specification languages have been defined that assist users to specify policies. The Chinese wall policy combines commercial discretion with legally enforceable mandatory controls. Unlike Bell-La Padula like policies, a user's permitted accesses are constrained by the history of his previous accesses. Several attempts have been made towards a single policy framework, which is able to investigate and enforce multiple security policies. Use logic-languages for the specification of authorizations for distributed systems [6]. Their proposal abstracts from low level authorization triples and adopts a high level specification language to achieve the need of expressiveness and flexibility.

2.2 Policy Composition and Analysis

Policy composition facilitates the sharing of protected data and resources among multiple parties in a controlled way [7]. It allows policies specified by more than one policy authors to be integrated to verify their compliance with the global requirements.

The algebra provided various type of operators for composing and restricting enterprise privacy policies like conjunction, disjunction and scoping together with its formal semantics. Security policies as access right matrices in terms of principals, typed objects and rights [8]. They define operations like Add, Or and Minus for combining and changing security policies.

- *Dominance check*: The effect of adding one policy to a group of existing ones. Policy A is dominated by policy group G if the adding of A does not affect the behavior of the system governed by G. Thus it helps to detect redundancy at the semantic level.
- *Coverage check*: Whether the specified policies have covered a certain range of input parameters.
- *Conflict check:* Detects conflict between two policies when they cannot be satisfied simultaneously.
- *Consistent priority assignment:* prioritizes policies by assigning an integer value to each policy. It is considered the primary method of resolving conflicts.

Among these tasks for policy analysis, lots of effort has been devoted to studying conflict detection and resolution techniques [9]. Proposed a logical language for the specification of authorizations. This language allows users to specify different kinds of security requirements, according to which access control decisions are to be made [10].

The logic representation also helps to perform conflict resolution and constraint checking. [11] reviews conflicts that may arise in a large-scale distributed system with role based management. Since management policies are specified in terms of domains, conflicts arise when there are overlapping between domains. Application specific conflicts can be resolved using meta-policies. [12] proposed a set of techniques to automatically discover policy anomalies and conflicts in centralized and distributed firewalls. Policy tree and state diagrams can be constructed to discover intra-firewall/inter-firewall anomalies and to determine the proper rule placement and ordering.

2.3 Policy Refinement

In policy-based security management, high-level security requirements need to be translated to low level rules, for which the syntax and semantics can be interpreted and implemented by individual enforcement points in order to make an appropriate and consistent decision upon receiving an access request. This process is often referred to policy refinement that remains one of the most ambitious goals in policy-based system management [13]. It fills the gap between policy specification and enforcement.

{Subject} can (or cannot) perform {Action} on {Target} if {Condition}.

It states that the subject is allowed or prohibited to perform an action on the target if certain condition is satisfied. Depending on the specific enforcement mechanism, subject (target) can be a simple identifier, a domain scope expression, a public key, etc [14]. The specified action field can be a high-level goal or a low-level operation, which emphasizes the needs for translating high-level policies into low-level mechanism rules for enforcement.

Existing work on policy composition focuses on the integration of policies using algebraic operations to produce compound rules. However, policy algebra goes beyond the semantic level when tied with policy distribution and enforcement. It allows policies to be rearranged in the network [15] and studies the enforcement effect of compound policies using algebraic operations.

Sometimes, policy analysis tasks cannot be applied directly to high-level policies as they carry less-detailed domain knowledge of the managed system. Therefore policy analysis must be interwoven with policy refinement to achieve desired results.

3 AN ALGEBRAIC FRAMEWORK FOR POLICY COMPOSITION AND DELEGATION

Security policy research largely focuses on the specification and management of access control requirements. The questions of how to understand the interactions between access control policies and how to enforce consistency in a policy-based system have not yet been adequately investigated. Moreover, existing policy composition and analysis solutions are mostly concerned with merging individual policy authors' security requirements in a controlled way. However, policy integration has another important but often neglected implication - that is to enable enforcement delegation in heterogeneous environments, where each device may incur different expense in terms of cost and risk for enforcing the same security policy.

Therefore, proposed an algebraic framework for policy composition, analysis and delegation, the first step towards a distributed policy management solution. Algebra defines mechanisms to reason about policy languages. It takes sets of policy rules as input and output, manipulates them, move them around, and combine them to understand the semantics of the policy language.

4 . SYSTEM OVERVIEW

A security policy consists of a set of information classes and constraints on flow of information. The constraints are specified by a specific type of logic called branching time temporal logic. A security policy is specified as a specific case of a regulation. The system to be regulated consists of agents which can execute actions on some objects. Each role is associated with a set of norms (permissions, obligations and prohibitions). An agent can play one or more roles. In this approach, regulation is specified using a logic based on SDL (Standard Deontic Logic). LaSCO (the Language for Security Constraints on Objects) is a language for specifying policy as a directed graph. The semantics of the language was represented by a first ordered logic.

A.ON{Event}IF{Condition}THEN {Action}

As it is well known, its semantics is as follows: if the event arises and the condition evaluates to true, the specified action is executed. In context, an event is the detection of an anomaly by the detection engine. A condition is specified on the attributes of the detected anomaly.

B. Anomaly Attributes

The anomaly detection mechanism provides its assessment of the anomaly using the anomaly attributes. Here identified two main categories for such attributes. The first category, referred to as contextual category, includes all attributes describing the context of the anomalous request such as user, role, source, and time. The second category, referred to as structural category, includes all attributes conveying information about the structure of the anomalous request.

C. Response Actions

Once a database request has been flagged off as anomalous, an action is executed by the response system to address the anomaly. A tainted request is simply marked as a potential suspicious request resulting in further monitoring of the user and possibly in the suspension or dropping of subsequent requests by the same user.

D. Policy Administration

The main issue in the administration of response policies is how to protect a policy from malicious modifications made by a DBA that has legitimate access rights to the policy object.

E. Policy Matching

Algorithms for finding the set of policies matching an anomaly. Such search is executed by matching the attributes of the anomaly assessment with the conditions in the policies. Base Policy Matching, Ordered Policy Matching and Response Action Selection.

A rule is defined as a set of criteria and an action to perform when a packet matches the criteria. The criteria of a rule consist of the elements direction, protocol, source IP, source port, destination IP and destination port. Therefore a complete rule may be defined by the ordered tuple direction, protocol, source IP, source port, destination IP, destination port, action. Each attribute can be defined as a range of values, which can be represented and analyzed as sets.

5. CONCLUSION

The management of network infrastructure in an enterprise is a complex and daunting affair. The concept of policy based management has help the administrator to manage the user actions, with proven validity as an intuitive and scalable way for administrators to keep large information systems under control, ensuring the continuous enforcement of domain directives. Here check how building a dependability management framework on a policy based core has indeed achieved to leverage the potential of this paradigm, applying it to a novel field. The proposed framework allows using the same abstract approach inherent to policy based solutions for managing also use Encrypt List is used for to translate the policies.

As future work, some extensions remain to be taken into account which would improve this framework considerably. The current configuration policies that govern the system should be extended to include also setting up in a similar manner the modules belonging to the framework; for example, the collection of sensors needed for a concrete operational plan, indicating the configuration for each of them. The history based approach is also use for to identify the anomalies in the given rule set.

6. REFERENCES

[1] A. Westerinen, “Terminology for policy-based management,” RFC-3198, Nov. 2001.

[2] Cataldo Basile and Antonio Lioy “Network-Level Access Control Policy Analysis and Transformation” *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 985–998, Aug. 2012.

[3] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, “Conflict classification and analysis of distributed firewall policies,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 10, pp. 2069–2084, Oct. 2005.

[4] C. Basile and A. Lioy, “Towards an algebraic approach to solve policy conflicts,” in *Proc.WOLFASI*, Turku, Finland, July 2004, pp. 319–338.

[5] C. Basile, A. Cappadonia, and A. Lioy, “Geometric interpretation of policy specification,” in *Proc. IEEE Policy*, New York, NY, Jun. 2008, pp. 78–81.

[6] S. Thanasegaran, Y. Yin, Y. Tateiwa, Y. Katayama, and N. Takahashi, “A topological approach to detect conflicts in firewall policies,” in *Proc. IEEE IPDPS*, Rome, Italy, May 2009, pp. 1–7.

[7] S. Ferraresi, S. Pesic, L. Trazza, and A. Baiocchi, “Automatic conflict analysis and resolution of traffic filtering policy for firewall and security gateway,” in *Proc. IEEE ICC*, Glasgow, Scotland, 2007, pp.–1310.

[8] M. Rezvani and R. Aryan, “Analyzing and resolving anomalies in firewall security policies based on propositional logic,” in *Proc. IEEE INMIC*, Islamabad, Pakistan, 2009, pp. 1–7.

[9] J. Zao, “Semantic model for IPSec policy interaction,” Internet Draft, Mar. 2000.

[10] Z. Fu, S. F. Wu, H. Huang, K. Loh, F. Gong, I. Baldine, and C. Xu, “IPSec/VPN security policy: Correctness, conflict detection and resolution,” in *Proc. IEEE Policy*, Bristol, U.K., 2001, pp. 39–56.

[11] Z. Li, X. Cui, and L. Chen, “Analysis and classification of IPSec security policy conflicts,” in *Proc. FCST*, Aizu, Japan, Nov. 2006, pp. 83–88.

[12] A.K.Bandara,E.C.Lupu, A.Russo, N. Dulay,M. Sloman, P. Flegkas, M. Charalambides, and G. Pavlou, “Policy refinement for IP differentiated services quality of service management,” *IEEE Trans. Netw. Service Manage.*, vol. 3, no. 2, pp. 2–13, Apr. 2006.

[13] J. D. Moffett and M. S. Sloman, “The representation of policies as system objects,” in *Proc. SIGOIS*, Atlanta, GA, 1991, pp. 171–184.

[14] J. D. Moffett and M. S. Sloman, “Policy hierarchies for distributed system

management," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 9, pp. 1404–1414, Nov. 1993.

[15] J. D.Moffett and M. S. Sloman, "Policy conflict analysis in distributed system management," *J. Org. Comput.*, vol. 4, no. 1, pp. 1–22, 1993.