ENTERPRISE WIDE CENTRALIZED APPLICATION LEVEL ACCESS CONTROL USING XACML

Riaz A. Shaikh
NIIT, Pakistan
55riaz@niit.edu.pk

Dr. Saeed Rajput
FAU, USA
srajput@fau.edu

Dr. S. M. H. Zaidi
NIIT, Pakistan
drzaidi@niit.edu.pk

Kashif Sharif
NIIT, Pakistan
kashif@niit.edu.pk

ABSTRACT

In traditional approach, enterprise-wide consistent security policy enforcement for applications is very difficult task. Therefore, industry is now moving towards new unified enterprise application security concept that consists of centralized authentication and authorization mechanism. The eXtensible Access Control Markup Language (XACML); an XML-based standard defined by OASIS, is most suitable choice which can support centralized, role based, context aware access control mechanism. It is designed to provide universal standard for writing authorization policies and access control request/response language for managing access to the resources. This paper includes a brief overview on XACML and discusses its benefits, limitations and a data flow process. We propose a new generic access control architecture that supports enterprise wide centralized application level access control mechanism using XACML. The other benefits which can be achieved through this architecture are, reduce administration cost and complexity, support of heterogeneous computing platforms, centralized monitoring system, automatic fail over, scalability and availability, open standard based solution and secure communication.

Keywords: Enterprise, Application, Access Control, XACML

1. INTRODUCTION

Conventional wisdom dictates embedding security code within each application so it has its own access control mechanism. This makes enterprise-wide consistent information policy enforcement across all applications tedious and a massive exercise in itself [1]. According to the CSI/FBI computer crime survey [2] the total annual losses reported in the 2003 were $141,496,560, in which unauthorized access by insider caused the financial loss of $4,278,205 and almost 52% of the security breaches coming from authorized users. New emerging regulations such as Gramm-Leach-Bliley act (GLB) [3], Health insurance portability and accountability act [4] etc, are enforcing pressure on the organizations to provide strong authentication, authorization and privacy mechanisms. Therefore, new unified enterprise application security concept is evolving in the industry, which consists of centralized authentication and authorization concepts [5].

In the classical research area of operating system security, several access control models [6] such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role Based Access Control (RBAC) are well understood. RBAC based security policies are more beneficial to the commercial sector than other policies [7]. The basic requirements for enterprise application security to provide safer access control system include;

- Centralized authorization mechanism,
- RBAC implementation,
- Granular access control, and
- Context aware access control.

Centralized authorization mechanism is needed for consistent policy enforcement and ease of management. RBAC is most commonly used scheme at enterprise level, and reduces the complexity of management further, and allows the security policies to be modified quickly, especially with role hierarchies are permitted. RBAC [6] provides policy neutral and flexible access control. It is possible to implement principle of least privilege and separation of duty easily as well. Granular access control allows organizations to provide better and new services to clients and users. Context aware access control is required to allow authenticated user to access resources at specific time and location [8]. The XACML (eXtensible Access Control Markup Language) is a new standard defined by OASIS and can be used to specify the policies to meet above requirements.

Section 2 of this paper contains an overview of XACML and covers its benefits, limitations, and its processing. Section 3 contains the proposed generic centralized access control mechanism. Section 4 contains benefits of proposed architecture. Section 5 presents the conclusions.

2. XACML OVERVIEW

The eXtensible Access Control Markup Language (XACML) is an XML-based standard define by the Organization for the Advancement of Structured Information Standards (OASIS). It is designed to provide universal standard for writing authorization policies and access control request/response language for managing access to the resources. The basic objectives of XACML are [9]

- Describing access control policies and their attributes in a portable and standard way.
- Providing mechanisms to support fine granular access control.

2.1 Basic Components

XACML defines four layers to access policy control [10]

1. Policy Administration Point (PAP) – It creates security policies or policy sets.
2. Policy Enforcement Point (PEP) – It performs access control by making decision requests and enforcing authorization decisions.
3. Policy Information Point (PIP) – It is an entity that serves as the source of attribute values, or the data required for policy evaluation.
4. Policy Decision Point (PDP) – It evaluates the applicable policy and renders an authorization decision.

2.2 Data Flow Process of XACML

The basic data flow model for XACML is defined in [11] is shown in figure 1. It involves following steps.

1. The policy or policy sets, which are written in PAP, are available to PDP.
2. Access request is intercepted by the PEP.
3. PEP forwards this request to the context handler in its native format, which optionally includes attributes of the sender, required resource, action and environment.
4. The context handler constructs the XACML request and sends it to PDP.
5. PDP request the additional information from context handler.
6. Context handler sends the request to PIP for required attributes.
7. PIP returns the required information to context handler.
8. Context handler optionally includes the resource in the context.
9. The context handler returns the requested attributes and optionally resource to PDP.
10. After evaluating the request it sends back a response or decision to the context handler.

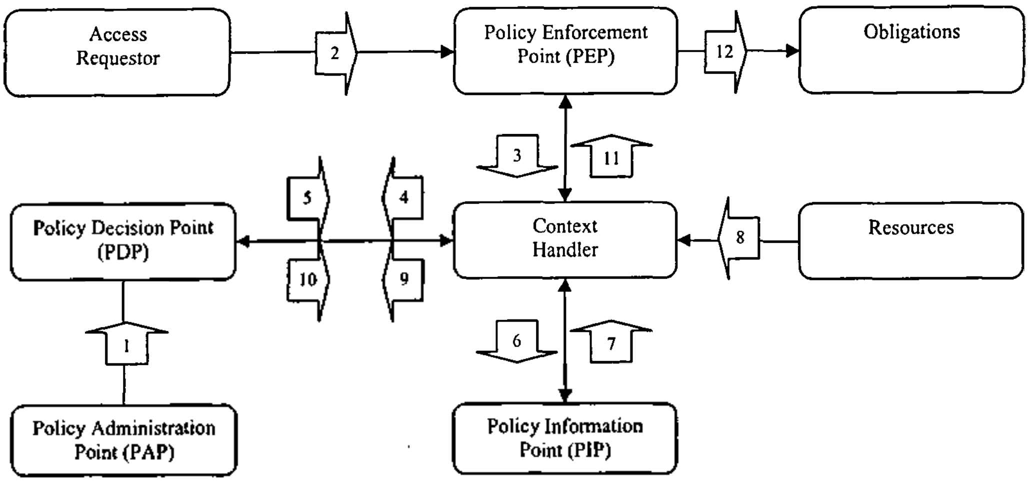
Access Requestor
2
Policy Enforcement Point (PEP)
12
Obligations
3
11
5
4
Policy Decision Point (PDP)
Context Handler
8
Resources
10
9
1
6
7
Policy Administration Point (PAP)
Policy Information Point (PIP)

Figure 1: Data Flow Diagram of XACML

11. Context handler converts this response to the native format of PEP and then sends this response/decision to the PEP.
12. The PEP fulfills the obligations. If access is permitted it gives access to the request other wise it denies access.

2.3 Benefits

XACML technology is useful in three areas [12] first in complex, interactive web services, 2nd in enterprise wide centralized security management and 3rd in Digital Right Management (DRM). Platform independence nature of XACML will allows enterprises to centrally manage the access control policies in a heterogeneous computing environment and enterprises can work together without having to align their diverse computing platforms (whether based on java, .NET or other distributed object technologies) [12]. XACML supports in DRM by defining how intellectual property can be accessed by individuals, automated agents or enterprises. Other major benefits of XACML over other access control policy languages are; it is a standard language, which replaces many application specific languages. It saves time for administrators who do not need to rewrite policies for different applications. It saves time and money for developers because once they write code policies in XACML form they can reuse it. It is flexible and extensible. It is use to support consistent policies on different resources. For large organizations it allows to refer one policy to another [13]. It is not only useful for centralized policy management environment but also it is good for distributed, decentralized or in grid computing environment [14]. XACML provides web service security building block along with Security Assertion Markup language (SAML), XML key Management Specification (XKMS) and Web-service security (WS-security) [12].

2.3 Limitations

There are number of limitations of XACML. For instance it does not provide explicit support to manage subject and object heterogeneity in a web service environments [15]. Subject heterogeneity creates problems in access control specification because users have various activity profiles such as characteristics or qualifications that may not be known priori. These activity profiles are essential for dynamically transferring of authenticated users from one web service to another. It also lacks conceptual level access control on objects. It assigns permission directly to users rather than assigning roles to abstract permissions. This violates the principles of scalability and manageability that motivates developers to use Role Based Access Control (RBAC) [15]. It does not support delegation model. It is difficult to express the idea of certificate because all the rules are composed in a single policy with one issuer or administrator. [16]. Service oriented architecture (SOA) in XML based access control technologies are still in infancy stage [17]. SOA is the combination of services which forms loose coupling to communicate with each other either in the form of simple data passing or involvement of two or more services organizing some activity.

2.4 Implementations of XACML

There are a number of different implementations of XACML such as Sun Microsystems and Parthenon. Sun Microsystems, Inc [18] has developed an open source implementation of OASIS XACML standard written in java. Parthenon software [19] has developed suite of policy products based on XACML. This suite contains policy tester, policy engine, and policy server.

2.6 XACML policy writer tools

Writing policies in XACML by hand is difficult and massive exercise especially for large organizations. There are number of tools are available for writing, managing and interpreting XML documents. Altova Corporation has developed an XACML policy writer tool [17] whose prototype is based on Altova Stylevision.29.

3. GENERIC CENTRALIZED ACCESS CONTROL MECHANISM (GCACM)

Our architecture is based on new unified approach which consists of centralized access control system illustrated in figure 2.

Core layer generally consist of authentication servers, policy servers, audit server etc. We are introducing two new components at core layer, the Policy Decision Module (PDM) and the Backup PDM (BPDM). Unified layer consist of policy enforcement modules (PEM) that are responsible of intercepting each access and response to the

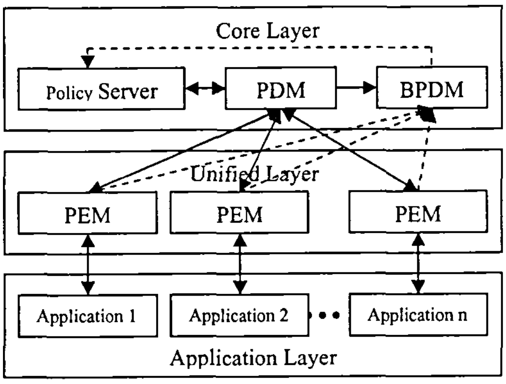

Figure 2: Unified Layer Concept

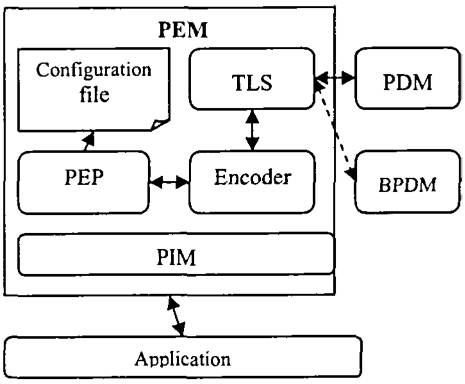

Figure 3: Policy Enforcement Module (PEM)

application and they are placed in front of each application. PEM evaluates the access control policy and enforces the decision. Application layer consist of critical applications. The main advantage of this type of architecture is that it separates policy enforcement from decision making, facilitating the centralized policy management.

PEM consist of Policy Enforcement Points (PEP), Encoder, Transport layer security (TLS) module, Protocol interpretation module (PIM) and configuration file. Logical diagram of PEM is shown in figure 3.

PEP is initialized at startup by reading configuration file which contain information about the PDM, BPDM locations. We are assuming that user is already authenticated and request coming from the authenticated user. When the request arrived at PEM then PEM is responsible for permitting or denying access to the application. For that purpose PEM first encode that request to XACML format with the help of Encoder module. Then this encoded request is forwarded to PDM in secure manner via Transport layer security (TLS). This request consists of attributes of the sender, required resource, action and environment. PIM is specific to each PEM implementations in order to support HTTP/S, SOAP, .Net, XML-RPC, CORBA, SMTP, TCP/IP etc capable applications.

PDM consist of Policy Decision Point (PDP), PDP-PBP Bridge, TLS, configuration file and Centralized Monitoring System (CMS). The basic architecture of PDM is shown in figure 4. The Core Module of PDM is PDP that will initialized at startup with by reading configuration file that contains the information about the BPDM and policy servers. There are two categories of policy servers; the XACML compatible policy servers and the proprietary based policy (PBP) servers. If the enterprise is using XACML compatible policy servers then PDP directly interacts with them server and obtains the access rights of the user other wise it will use PDP-PBP Bridge to interact with proprietary based policy servers. The PDP-PBP Bridge is responsible for establishing connection and conversion of request into native format of the proprietary based policy servers. When PDP gets access rights of the users, it will evaluate the request according to the access rights and send back response or decision to PEM. When the PEM gets the response from PDM it will enforces this decision by permitting or denying access to the user. CMS is used to monitors critical operational information that facilitates the system administrators to respond rapidly to problems.

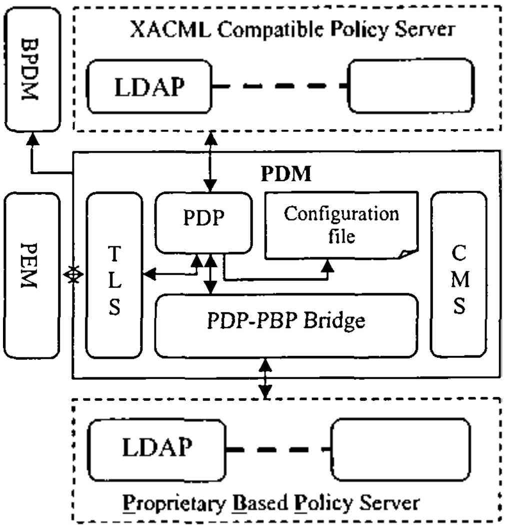
BPDM
XACML Compatible Policy Server
LDAP
PDM
PEM
PDP
Configuration file
TLS
CMS
PDP-PBP Bridge
LDAP
Proprietary Based Policy Server

Figure 4: Policy Decision Module (PDM)

The BPDM is a backup for PDM. PDM periodically sends "keep alive" messages to BPDM. If, within a specific time BPDM does not receive a keep alive message from PDM, it will consider it dead and take a charge of PDM. This thing will increase the reliability and availability.

4. BENEFITS OF GCACM

Centralized policy management: The major benefit of this architecture is centralized policy management which will ensure the enterprise wide consistent information policy enforcement across all applications.

Reduced administration cost and complexity: Centralized policy management helps system administrators to quickly maintain and update the policies. This thing minimizes the administration cost and complexity.

Support of heterogeneous Computing Platforms: PEM is independent of any applications and it can be easily integrated with any application which is based on SOAP, XML-RPC, CORBA, TCP/IP etc.

Centralized monitoring system enables system administrator to quickly respond to any problem by monitoring critical operational information. It will also help to determine weak areas and spot attempted security breaches.

Automatic Fail Over: mechanism increase the reliability, performance and availability of system across 24x7.

Scalability: Replication of PDM, automatic fails over and independent of any application, increases scalability of a system. Increase in a number of applications or number of users does not create any affect on this architecture.

Open, standard based solution: This architecture incorporates different industry standards like XACML, TLS which will help in interoperability among different vendors, products or implementations. Open, standard based security solution is more preferable by organizations as compare to proprietary based solutions.

Secure Communication: Transport layer security provides a secure communication between the different components of architecture by using different encryption algorithms and it will also ensure that the messages are coming from the authenticated source not from spoof source.

5. CONCLUSION

Centralized policy management and distributed enforcement is the major benefit of our proposed architecture. This architecture provides interoperability with XACML, which provides universal standard for writing authorization policies for managing access to the resources. It also provides access control request/response language. The other benefits of this architecture are, reduces administration cost and complexity, supports of heterogeneous computing platforms, centralized monitoring system, automatic fail over, scalability and availability, open standard based solution and secure communication.

6. REFERENCE

[1] Gilson Wilson, Ullas O. Tharakan, "Unified Security Framework", *proceedings of the 1st international symposium on Information and communication technologies*, Dublin, Ireland ,Sep. 2003, pp. 500-505

[2]Lawrence A. Gordon, Martin P. Loeb, "Computer Security Issues and Trends", *9th annual CSI/FBI Computer Crime and Security Survey, Computer Security Institute*, 2004

[3] Gramm-Leach-Bliley (GLB) Act, "Standards for safeguarding customer information", *FTC*, May 23, 2002 URL: http://www.ftc.gov/os/2002/05/67fr36585.pdf (Accessed: 11 Dec, 2004)

[4] Health insurance portability and accountability act of 1996, *Public law, 104th congress*, URL: http://aspe.os.dhhs.gov/admnsimp/pl104191.html (Accessed: 11 Dec, 2004)

[5] Saeed Rajput, and Basit Hussain, "Application Defense: Next Generation of Unified Enterprise Security", *International Workshop on Frontiers of Information*, Islamabad, Pakistan December 23 - 24, 2003

[6] James B. D. Joshi, Walid G. Aref, Arif Ghafoor, and Eugene H. Spafford, "Security models for web-based applications" , *Communications of the ACM*, Vol. 44, No. 2, 2001, pp. 38-44

[7] Axel Kern, "Advanced Features for Enterprise-Wide Role-Based Access Control", *proceedings of 18th Annual Computer Security Applications Conference*, 2002, pp. 333-342

[8] IBM Tivoli Software, "Securing the Enterprise", Oct, 2002, URL: http://www.orb-data.com/cgi-bin/launch.cgi?page=Securing_the_enterprise (Accessed: 7 Feb 2005)

[9] Manish Verma, "The objectives, architecture, and basic concepts of eXtensible Access Control Markup Language", 18 Oct, 2004. URL: http://www-128.ibm.com/developerworks/xml/library/x-xacml/

[10] Michael W Armstrong, "An Introduction to XACML", *GIAC practical repository*, SANS Inst, 29 Jun, 2003

[11] Tim Moses and Entrust, "eXtensible Access Control Markup Language 2 (XACML) Version 2.0", *OASIS Committee Draft 02*, 30 Sep, 2004

[12] Ray Wagner, "XACML Will Help Enterprises in Three Areas", *Gartner, Inc. Research*, 21 Feb. 2003

[13] Sun Microsystems, Inc., "Sun's XACML implementation Programmer's Guide for Version 1.2" URL: http://sunxacml.sourceforge.net/guide.html (last update: 11 July 2004)

[14] Markus Lorch, Seth Proctor, Rebekah Lepro, Dennis Kafura, and Sumit Shah, "First experiences using XACML for access control in distributed systems", *Proceedings of the 2003 ACM workshop on XML security*, Fairfax, Virginia, 2003, pp. 25-37

[15] Rafae Bhatti, Elisa Bertino, Arif Ghafoor, James B.D., Joshi, "XML based specification for web services document security", *IEEE Computer Society*, April 2004, pp. 41-49

[16] G. Navarro, B.S. Firozabadi, E. Rissanen, and J. Borrell, "Constrained Delegation in XML-based Access Control and Digital Rights Management Standards", *proceeding of 6th annual Communication, Network, and Information. Security(CNIS)*, 2003, pp. 271-276

[17] David Staggs, "Evaluation of XML Technologies as Applied to Access Control", Science Application Int. Corp., 13 Sep, 2004

[18] Sun Microsystems Inc., "Sun's XACML implementation", URL: http://sunxacml.sourceforge.net (last update 7 Jan, 2005)

[19] Parthenon Computing Ltd, "XACML Policy Products", URL: http://www.parthenoncomputing.com/products/xacml/index.html (Accessed: 2 Feb, 2005)