EAM시스템 보안을 위한 XACML과 PMI 상호운용에 관한 연구

박제원[O], 정성우, 이남용
숭실대학교 대학원 컴퓨터학과
{ kkkjw22 [O], jsw9999 }@hanmail.net , Nylee@Computing.ssu.ac.kr

A Study of the interoperability of XACML and PMI to enhance the security on EAM system

Jaewon Park[O] , Sungwoo Jeong , Namyong Lee
Department of Graduate School, Soongsil University

요 약

EAM(Enterprise Access Management)은 SSO(Single Sign On)와 사용자 역할기반의 세분화된 접근관리를 제공하는 적극적 시스템 인증, 제어관리 솔루션으로 EAM 도입을 위해서는 중요한 자원들의 안전한 보관 관리가 필수적이다. 이를 위해 각 기업이나 그룹에서는 XML을 기반으로 한 여러 보안기술을 도입하고 있지만, 현재 XML 정보 보안 기술을 중심으로 한 EAM시스템의 구현은 그 기반이 미미한 실정이며, 표준에 따른 시스템의 고려도 미약한 실정이다. 특히 XML 정보보호기술과 기존 보안기술인 공개키기반구조(Public Key Infrastructure, PKI), 권한관리구조 PMI(Privilege Management Infrastructure)등과 같은 인증기술과의 상호연동기술에 관한 연구가 부족한 상태이다. 이에 본 논문은 표준 XML 정보 보호기술 중 대표적인 기술인 XACML(eXtensible Access Control Markup Language)과 PMI를 연동한 안전한 EAM 통합 접근 시스템 구축 방안에 대해 연구하였다.

1. 서 론

EAM은 현재의 기업에서 부각된 문제에 대한 요구에서 출발한 분산되어 운영되고 있는 자원과 사용자에 대한 통합을 통한 일관된 관리체계를 구축하는 보안 솔루션이다. 하지만 기업의 전산 환경이 확장되고 다양한 시스템과 다양한 사용자를 관리 해야 하는 상황이 발생하고, EAI(Enterprise Application Integration), EIP(Enterprise Information Portal)등의 통합이라는 개념과 맞춤 보안이라는 인식이 점점 확산됨에 따라 이러한 모든 상황에서 안전하게 사용될 수 있는 보안기술이 필요하게 되었다. 이에 따른 EAM시스템에 적합한 XML보안 기술은 웹 서비스 표준화 기구인 OASIS와 W3C의 XML 암호화, 전자서명, 접근제어 등의 XML기반 보안기술을 들 수 있고, PKI공개키 방식과 이를 확장한 PMI 속성인증기술 등을 사용하여 문제 해결에 접근할 수 있다. 본 논문에서는 XML 기반 보안기술 중의 하나인 OASIS의 XACML과 PMI와의 연동을 연구하고 이를 기반으로 한 EAM의 적용 방안을 연구했다.

2. 관련 연구

2.1 XACML

EAM 시스템은 XML을 전자문서포맷으로 사용하는데 이에 따라 지속적으로 증가하는 XML 기반의 데이터들에 대한 접근 제어도 필수 보안 요구 사항이 되었다. 다양한 접근 제어 기법 중 OASIS의 XML정보보호 표준중의 하나인 XACML은 접근제어 정책을 통해 보안이 요구되는 자원에 대해 미세한 접근 제어 서비스를 제공 할 수 있는 XML 기반의 언어이다. XACML의 정의에 따라 각각의 사용자 별 XML 데이터 접근 정책을 수립하고 적용 할 수 있다. 접근에 대한 허가 또는 거부와 같은 단순한 접근 제어를 하는 것이 아니라 보다 미세한 접근 제어 모델을 제공한다. [1],[2]

XACML은 XML로 기술된 정책 언어와 접근제어 결정 요구/응답 언어로 구성되어 있다. 정책 언어는 규칙, 정책 및 정책 셋 등에 관한 통상적인 접근제어 요구사항들에 대해 기술하고 있으며 함수들, 데이터 타입들 및 조합 논리 등에 대해서도 정의하고 있다. XACML은 2003년 7월에 XACML v1.1 표준이 완성됐고, 현재 XACML v2.0 개발이 진행 중이다.[5]

2.2 PMI

PMI는 인증서 구조에 사용자에 대한 속성 정보를 제공하여 권한 관리가 가능하도록 하는 속성 인증서 기술과 속성 인증서를 발급, 저장, 유통을 제어하는 기반구조이다. 속성 인증서가 정보보호 메커니즘으로 활용되기 위해서는 속성 인증서의 발급, 저장, 유통이

속성 인증서 생성 기관, 속성 인증서 소유주, 응용 서비스 시스템 등에서 원활히 동작할 수 있어야 한다. PMI는 속성 인증서의 발급, 저장, 유통, 검증 등을 포괄하는 권한관리 기반구조이다. 여기에서 권한은 속성과 같은 의미이다. 따라서 PMI를 구성하는 방법에 따라, 속성 인증서의 활용 방식과 응용 서비스 환경이 영향을 받게 된다.[8]

속성 인증서를 분배하는 방식은 pull 모델과 push모델 두 가지가 사용된다. 본 논문에서 사용된 pull 모델은 속성 인증서가 생성되었을 때 디렉토리에 속성 인증서를 게시하는 방식이다. 따라서 속성 인증서를 사용하는 응용 서비스는 속성 인증서가 필요할 때 디렉토리에서 인증서를 검색하여 사용한다. 클라이언트 또는 클라이언트-서버 프로토콜의 변경 없이 구현될 수 있다는 장점을 가지고 있으며 클라이언트의 권한이 서버 도메인 내에서 할당되어야만 하는 경우에 적합한 모델이다.

3. EAM 보안을 위한 XACML과 PMI 상호 운용

본 논문에서는 EAM을 구성하는 사용자 인증 및 자원 접근 관리에 대한 보안기술로서 PKI의 단점을 보안한 PMI와 XML기반 접근제어기술 표준인 XACML을 상호 연동한 시스템을 제시한다.

3.1 시스템 아키텍처

[그림 1]은 본 논문에서 제안하는 시스템 아키텍처이다.

Privilege Creator는 X.509 속성인증서의 정보와 XACML규칙에 따라 인코딩된 권한정보를 생성하여 사용자의 권한 속성을 정의한다. 권한 속성 구조는 Issuer, Holder, Lifetime, Privilege Statements, Signature로 구성되어 있다. 이중 Privilege Statements는 XACML 정책 언어의 규칙대로 인코딩되어 있으며 그 외의 요소들은 X.509 속성 인증서의 정보를 사용하였다.[3],[4]

Policy Creator는 XACML표현의 문법적인 복잡성으로 인해 필요한데 자원 허가에 대한 질의를 분석하여 미리 접근제어 정책을 XACML 형식으로 작성한다.

요청자가 특정 자원에 대해 어떤 동작을 원할 때 PEP(Policy Enforce Point)에게 자원 허가에 대한 질의를 한다(1 단계). PEP는 PDP(Policy Decision Point)에 개체, 자원, 동작의 속성들을 포함하는 자원 허가에 대한 질의를 한다(2 단계). PDP는 적용 가능한 정책을 찾고 필요한 속성들을 검색하고 평가한다(3 단계). PDP는 권한 부여 결정에 대한 응답을 PEP에 보낸다(4 단계) PEP는 의무조항을 수행한다. PEP가 의무조항을 이해할 수 없거나 수행하지 못하면 접근을 거부해야 한다. 그리고 자원에 대한 접근이 허용되면 PEP는 자원에 대한 접근을 허용한다. 그렇지 않으면, 접근을 거부해야 한다(5 단계). 서비스에 대한 응답이 PEP로 전달되고(6 단계) PEP는 개체에 응답을 전달한다(7 단계).

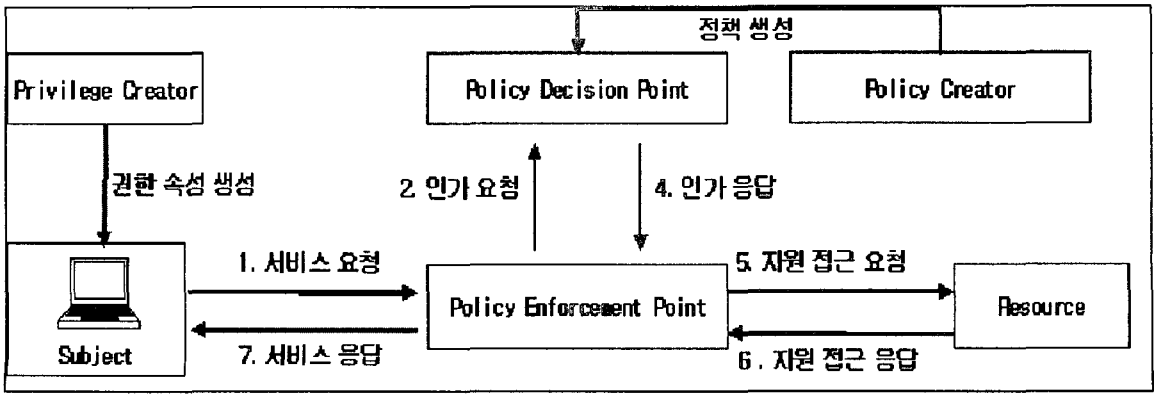
[그림 2]는 파일 접근을 위해 권한 속성 구조의 Privilege Statements를 XACML규칙에 의해 인코드한 예제이다.

3.2 결론

본 논문에서 제시한 XACML과 PMI의 연동 시스템을 사용함으로써 소유 정보 변경에 따른 권한 전달과 접근 제어 정책 생성의 재사용이 가능해졌으며 이에 따라 기존의 파싱과 개발 코드의 재사용이 가능해졌다. 또한 보다 동적인 정책 생성이 XACML을 사용함으로써 가능해졌고 다양한 접근 제어 시나리오를 적용하는데 있어 기존의 다른 언어로 만든 컴포넌트와 쉽게 연동이 가능하다.

본 시스템에서는 접근 제어 정보 전달을 위해 PMI에서 사용한 X.509 속성 인증서 버전3을 사용하였다. 여기서 버전 2와 3의 차이는 확장필드의 유무에 따라 다른데, 이 필드는 키와 이름 이외에도 부가적인 정보를 실을 수 있도록 보다 나은 유연성을 부여한다. XACML은 X.500의 이름들을 지원하기 때문에 X.509 인증서에 의해 식별되는 엔티티들과 접근 정책 규칙을 직접 연결할 수 있다.

XACML로 변경함에 따른 단점은 접근 정책과 권한 전달시에 XML 인코딩에 따른 오버헤드가 발생하며 XML문서의 크기가 증가하는 것이다.



<그림 1> 시스템 아키텍처

```
<Rule RuleId="File-Privilege-Rule" Effect="Permit">
  <Target>
    <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:x500Name-equal">
            <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name">
                CN=Park jaewon ,OU=Soongsil Uni User,OU=Class 2,O=setest,C=KOREA
            </AttributeValue>
           <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name" />
         </SubjectMatch>
        </Subject>
       </Subjects>
     <Resources>
       <Resource>
         <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                 setestftp://test.setest /data/eamtest/results.dat
          </AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
         </ResourceMatch>
        </Resource>
       </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                    Read
            </AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
           </ActionMatch>
          </Action>
         </Actions>
        </Target>
</Rule>
```

[그림 2] XACML규칙에 의해 인코드한 Privilege Statements

4. 향후 연구

본 논문에서는 EAM 솔루션으로써 OASIS 인증 표준인 XACML과 기존 정보보호 기술인 PMI와의 연동을 연구 하여 안전한 EAM통합 접근시스템을 제시하였다. 제시된 논문을 통한 향후 연구로는 XACML에 SAML의 인증기술을 추가하여 XML보안기술 부분을 보완 할 것이며, XKMS, XML Encryption 등 다른 표준 XML 정보보호기술과 사용하기 위한 프로파일 개발과 여러 전송 프로토콜들과 사용하기 위한 바인딩 기술에 대한 연구 또한 추가할 예정이다. 마지막으로 분산환경 등에서의 각각의 PDP 정책에 대한 생성, 분류, 평가 및 집행에 관한 효율적인 정책관리 방법도 차후 과제로 실행할 예정이다.[6],[7]

5. 참고문헌

[1]Markus Lorch, Seth Proctor, Rebekah Lepro, First Experiences Using XACML for Access Control in Distributed Systems, ACM Workshop on XML Security, 2003

[2]OASIS, eXtensible Access Control Markup Language Version 1.0, 2003

[3]D. Chadwick and A. Otenko, "The Permis X.509 Role Based Privilege Management Infrastructure", ACM Press, 2002

[4]Markus Lorch, David Adams, Dennis Kafura, Madhu Koneni, Anand Rathi, Sumit Shah "The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments", communicated to the 4th Ind. Workshop on Grid Computing - Grid 2003

[5]Tim Moses, Anne Anderson, Seth Proctor, and Simon Godik, "XACML Profile for Web Services", OASIS TC Working Draft, September 29th, 2003

[6]Sidharth Nazareth, "SPADE: SPKI/SDSI for AttributeRelease Policies in a Distributed Environment", Dept of Computer Science, Dartmouth College Technical Report TR2003-453, May 30, 2003

[7]OASIS Registry Technial Committee, "OASIS/ebXML Registry Services Specification v2.0", April 2002 Distributed Systems and Networks

[8] 진승헌, 최대선, 조영섭, 윤이중, 속성인증기술과 PMI, 정보보호학회지, 2000