Um Modelo de Controle de Acesso para Conteúdos Digitais

Valerio Rosset, Carla Merkle Westphall

INE - LRG (Laboratório de Redes e Gerência)
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brasil
{valerio, carla}@lrg.ufsc.br

Abstract. *Nowadays an increasing amount of information are being available using XML (eXtensible Markup Language). This information can be, for example, corporative information or business-to-business information, imposing the necessity of providing security in the environment. This paper presents an access control model for XML documents, describing the subjects, objects and the authorization rules. The prototype implemented validates the model and provides access control for web applications.*

Resumo. *Atualmente um crescente volume de informações estão sendo disponibilizadas na* web *através do uso do XML (eXtensible Markup Language). Essas informações podem ser, por exemplo, corporativas ou referentes a B2B (business-to-business), impondo a necessidade de fornecer a segurança no ambiente. Este trabalho apresenta um modelo para controlar o acesso a documentos XML, descrevendo os sujeitos, objetos e as regras de autorização. O protótipo implementado valida o modelo e provê controle de acesso para aplicações web.*

Palavras Chave – Políticas de Segurança, XML, Controle de Acesso.

1. Introdução

Um crescente volume de informações e conteúdos estão sendo representados de maneira estruturada e semi-estruturada tanto em *intranets* corporativas como também na Internet. As informações disponibilizadas através do uso do HTML são caracterizadas como semi-estruturadas, por que o HTML não fornece uma separação clara entre a estrutura e *layout* de um documento [Damiani 2002]. A estruturação de informações pode ser atribuída ao uso de linguagens de marcação como o XML (*eXtensible Markup Language*) [Bray 2000]. O XML vem se tornando um padrão para a integração de dados entre diferentes aplicações *web*. Neste contexto, garantir a segurança das informações disponibilizadas através do XML torna-se indispensável.

A integridade, confidencialidade e auditoria são algumas características de segurança. A definição e uso de uma linguagem de controle de acesso a informações (*Access Control Language*), através de um modelo e regras definidas por políticas de segurança, tornam possível o uso e a integração dessas características de segurança em conteúdos digitais. Através de um mecanismo de controle de acesso é possível definir e autorizar as ações de usuários sobre documentos ou partes de documentos XML como, por exemplo, ler ou alterar qualquer parte de um documento.

Assim é necessário definir um modelo e políticas de segurança para o acesso restrito a documentos XML que integre todas as principais características de segurança e permita a interação com aplicações w*eb*. Este trabalho está inserido neste contexto,

apresentando um modelo para controlar o acesso a documentos XML, evoluindo alguns aspectos não tratados ainda na literatura.

Este documento está organizado da seguinte maneira. Na seção 2 são apresentados os conceitos preliminares referentes a XML e a segurança em documentos XML. O modelo de controle de acesso é tratado na seção 3. A seção 4 apresenta os resultados da implementação e finaliza com a seção 5 onde são apresentadas as considerações finais.

2. Conceitos Preliminares

2.1 XML (*eXtensible Markup Language*)

XML [Damiani 2000] é uma linguagem de marcação para descrição de informação semi-estruturada. Um documento XML é formado por uma seqüência de elementos aninhados, cada um delimitado por um par de *tags*, um para o início e outro para o fim (e.g., <Exame> e </Exame>) ou por uma *tag* vazia. Documentos XML podem ser classificados quanto a formatação como bem-formados (*well-formed*) e válidos (*valid*).

Um documento XML é bem formado quando ele obedece à sintaxe do XML, não possuindo *tags* vazias ou quando um elemento com marcação de início não tem marcação de final. Um documento bem formado ou bem escrito é válido quando ele obedece à estrutura determinada em sua *Document Type Definition* (DTD). Uma DTD é um arquivo que contém a definição dos tipos e elementos que fazem parte do documento XML. A figura 1 mostra um exemplo de uma DTD.

Uma DTD pode incluir declarações para elementos, atributos, entidades e comentários. Os elementos [Damiani 2000] são os componentes mais importantes de um documento XML. A declaração de elementos em uma DTD especifica o nome dos elementos e seu conteúdo. Ela também pode descrever sub elementos na sua estrutura utilizando uma sintaxe formada de símbolos chamados de indicadores de ocorrência e conectores. O símbolo "*" indica nenhuma ou mais ocorrências, já o "+" indica uma ou mais ocorrências, o "?" indica nenhuma ou uma ocorrência e quando não houver nenhum indicador assume-se apenas uma ocorrência daquele elemento em todo o documento.

```
<!-- Documento XML  -->
<?xml version="1.0" encoding="UTF-8"?>
<Exame>
<Resultado ID="Rosset">
<Sangue>
<PLaquetas>0.5</PLaquetas>
<Hemoglobina>0.12</Hemoglobina>
<HIV>negativo</HIV>
</Sangue>
</Resultado>
</Exame>
```

```
<!-- DTD  -->
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Exame (Resultado*)>
<!ELEMENT Resultado (Sangue?)>
<!ATTLIST Resultado ID ID #REQUIRED>
<!ELEMENT Sangue (PLaquetas?, HIV?,
                  Hemoglobina?)>
<!ELEMENT Hemoglobina (#PCDATA)>
<!ELEMENT HIV (#PCDATA)>
<!ELEMENT PLaquetas (#PCDATA)>
```

Figura 1 – Exemplo de um documento XML e sua respectiva DTD

Os atributos representam as propriedades dos elementos. Declarações de atributos especificam cada atributo de um elemento bem como nome, tipo e um valor padrão. Entidades e comentários são importantes para a estrutura física de um documento XML.

A DTD não é a única maneira para definir um documento em XML. Para isso, também pode ser utilizado o XML *Schema* que permite a especificação de elementos globais e locais bem como todas as características do DTD. O XML *Schema* torna-se

mais atraente que o DTD quando os documentos XML tornam-se maiores e mais complexos. Da mesma maneira podem existir casos em que os documentos são menores e possuem uma estrutura mais simples, neste caso é aconselhável o uso da DTD.

Sendo assim a escrita de um documento XML deve seguir as determinações estabelecidas por uma DTD ou *Schema* referente a cada documento, para que ele possa ser considerado válido e conseqüentemente bem formado. A figura 1 demonstra um documento XML e sua respectiva DTD.

2.2 Acesso a documentos XML

O acesso a um documento ou partes de um documento XML pode ser feito através de uma aplicação que permitirá a leitura ou escrita de documentos XML. Essa aplicação faz uso de uma API em combinação com um *parser* XML, que permite ao usuário manipular dados em nós de elementos de um documento.

As duas APIs mais utilizadas para a manipulação de documentos XML são SAX[1] e o DOM[2] . O SAX (*Simple API for XML*) foi desenvolvido por um grupo de desenvolvedores em XML. O DOM (*Document Object Model*) foi desenvolvido e padronizado pelo W3C. A principal diferença entre esses dois conjuntos de APIs está no escopo, procedência e estilo de programação.

Quando o usuário, através de uma aplicação, requisita um documento ou parte de um documento XML, a API de manipulação aciona o *parser* que se encarrega de validar o documento requisitado, através da DTD ou *Schema* do documento. Validado o documento a API envia para a aplicação todos os dados requisitados, sejam eles um conjunto de nós ou um único elemento do documento XML. O acesso a partes de documentos como, por exemplo, de um determinado elemento qualquer é feito pelo XPath. O XPath [Clark 1999] é uma linguagem para endereçamento de partes de um documento XML. O uso do XPath facilita o acesso direto aos elementos de um documento XML, não necessitando a navegação no documento.

Finalmente de posse dos dados a aplicação pode ainda convertê-los em um formato desejado pelo usuário, através de uma linguagem de transformação de documentos XML designada como XSLT [Clark 1999].

2.3 Segurança em XML

Existe uma série de padrões para suprir requisitos de segurança de documentos XML. Esses padrões de segurança definem vocabulários em XML para representar informações seguras, usando tecnologias como, por exemplo, XML *Schema*, DTD e XPath. Os padrões de segurança permitem que a segurança seja aplicada em documentos XML, elementos do documento bem como seu conteúdo.

Os principais padrões de Segurança em XML são:

- *XML Digital Signature*[3], para integridade e assinaturas;
- *XML Encryption*[3], para confidencialidade;
- *XML Key Management* (XKMS)[3], para gerenciamento de chaves;
- *Security Assertion Markup Language* (SAML)[4], para autenticação e autorização;

[1] http://www.megginson.com/SAX
[2] http://www.w3.org/DOM
[3] http://www.w3.org
[4] http://www.oasis-open.org

- *XML Access Control Markup Language* (XACML)[4], para estabelecer como definir as políticas para o controle de acesso.

No modelo de controle de acesso proposto ainda não são utilizados esses padrões de segurança, o acesso é determinado por um conjunto de regras com sintaxe definida, conforme será apresentado na seção 3.

3. O Modelo de Controle de Acesso

O modelo de controle de acesso proposto neste trabalho estabelece um método de controle para escrita em documentos XML. Sendo assim, fica claro que primeiramente deve-se desenvolver também um controle para leitura desses documentos. Vários trabalhos foram realizados para definir um controle de acesso para leitura em documentos XML, entre eles destacam-se o [Gabillon 2001] e [Damiani 2002]. O modelo de acesso para leitura é inspirado nesses dois trabalhos, aproveitando algumas de suas características, para melhor definição do controle de acesso.

O desenvolvimento de um modelo de controle de acesso requer a definição de sujeitos, objetos e regras de autorização que especificam o controle de acesso.

3.1 Sujeitos

Um sujeito pode ser um usuário. Neste modelo cada sujeito ou usuário é identificado por um ID que pode ser representado, por exemplo, um nome, *login* ou um número IP. Cada usuário pode pertencer a um ou vários grupos de usuários. Os usuários e grupos de usuários são armazenados em um documento XML que obedece à hierarquia de usuários definida por [Gabillon 2001] no modelo *XML Subject Sheet* (XSS). A figura 2 apresenta um exemplo de um documento XML de usuários.

```
<sujeitos> <usuarios> <membro Id="Rosset"> <nome>Valerio Rosset</nome> </membro>
                       <membro Id="Souza">  <nome>João Souza </nome>    </membro>
           </usuarios>
           <grupos> <Pacientes> <membro Idref="Rosset"/>  </Pacientes>
                    <Doutores>  <membro Idref="Souza"/>   </Doutores>
            </grupos>
</sujeitos>
```

Figura 2 – Exemplo de um *XML Subject Sheet* (XSS)

O documento XSS representado na figura 2 apresenta um exemplo onde é definida a hierarquia entre dois tipos de usuários. São definidos dois grupos: Pacientes e Doutores. Cada grupo pode conter subgrupos, dependendo da necessidade de cada usuário.

A seleção de informações de usuários é feita através do seu *path* (caminho) específico, sendo assim pode–se ter os seguintes exemplos de caminhos:

- *path* (Usuários/) seleciona as informações de todos os usuários.
- *path* (Usuários/membro[@ID=' Rosset']) seleciona as informações referentes ao usuário Valério Rosset.
- *path* (Grupos/Doutores/* [nome!= 'Souza']) seleciona todas as informações de todos os usuários exceto as referentes ao usuário João Souza.

3.2 Objetos

O objeto é um recurso que o sujeito ou usuário tem acesso em determinado documento XML. Um objeto pode ser considerado um nó da árvore de estrutura de um documento XML. Na figura 1, por exemplo, a estrutura representada pelo nó (Sangue) é o objeto o qual o usuário (ID=Rosset) possui acesso.

3.3 Regras de Autorização

Uma vez identificados os Sujeitos e Objetos, torna-se necessária a definição do modelo de regras que determina quais os tipos de acesso que os sujeitos terão a cada objeto.

O modelo de regras de autorização foi definido baseado na proposta de [Gabillon 2001] e [Damiani 2002], onde o modelo proposto por [Gabillon 2001] é formado por quatro tuplas de autorização a tupla *Subjects* (sujeito), *Objects* (objeto), *Access* (negado ou liberado) *e Priority* (prioridade).

No modelo proposto em [Damiani 2002] são utilizadas cinco tuplas para a definição das regras de autorização (*Subject, Object, Action, Sign, Type*), este modelo se diferencia do modelo proposto em [Gabillon 2001] em três aspectos. Primeiro pela utilização da tupla *Action* que define o tipo de ação que o Sujeito poderá executar sobre o Objeto, como por exemplo, Leitura ou Escrita. Segundo, pela utilização de uma tupla para definição do tipo de acesso que um sujeito pode ter a um determinado Objeto, no que diz respeito aos nós da arvore do documento XML, determinando se o acesso ao Objeto, por exemplo, é recursivo ou local. E finalmente a terceira diferença é que ele não utiliza a tupla para a definição de prioridade. A tupla *Sign* equivale a tupla Access do modelo proposto em [Damiani 2002].

O modelo de regras autorização proposto neste trabalho é formado por um conjunto de seis tuplas, são elas:

- Sujeito: Define a qual o usuário ou grupo de usuários a regra se refere.
- Objeto: Define o que o Sujeito poderá ou não acessar.
- Acesso: Define se o Sujeito possui ou não permissão (Permitido ou Negado)
- Tipo de Acesso: Define o tipo de acesso ao Objeto (Leitura ou Escrita)
- Prioridade: Indica qual a prioridade da regra (0 ou 1)
- Modo de Acesso: Indica qual o modo de acesso ao Objeto (Local ou Recursivo)

O conjunto de regras que formam as políticas de segurança para o controle de acesso é armazenado em um documento XML denominado *XML Policies Sheet* (XPS).

```
<XPS ArquivoUsuarios="Usuarios.xml" PoliticaPadrao="open">
<! -- Regra 1  -- >
       <Regra Sujeito="/sujeitos/grupos//Pacientes"
       Objeto="/Exame/Resultado" Acesso="Deny" TipoAcesso="Escrita"
       ModoAcesso="Local" Prioridade="0"/>
<! – Regra 2 -- >
       <Regra Sujeito="/sujeitos/grupos//Doutores"
       Objeto="/Exame/Resultado" Acesso="Grant" TipoAcesso="Escrita"
       ModoAcesso="Recursivo" Prioridade="0"/>
</XPS>
```

Figura 3 – Regras de autorização no documento XPS

Na figura 3 é apresentado um exemplo da implementação das regras de autorização no documento XPS, escrito e validado por sua DTD. A primeira regra determina que todos os pacientes estão impedidos de escrever ou alterar o conteúdo de

todos os resultados. A segunda regra determina que todos os usuários do grupo doutores têm acesso permitido para escrita de resultados.

3.4 Algoritmo de Leitura e Escrita

Quando um usuário qualquer requer uma determinada informação, contida em um documento XML, [Gabillon 2001] deve ser permitida a ele apenas a visão do conteúdo que seja compatível com seus direitos. Isso vale tanto para a escrita como para a leitura em documentos. Para que o usuário tenha a visão apenas do que é determinado para ele, pelas regras de autorização, é necessária a utilização de um algoritmo que deve coletar as informações referentes ao usuário em questão.

```
1. Recebe (Sujeito, Documento Alvo)
2. Verifica Grupo (Sujeito)
3. Verifica Políticas (Grupo, Sujeito, Documento de Políticas)
Visao {lista de nós vazia}
4. N ← Recebe a lista de nós documento alvo
       referente aos objetos definidos nas regras
       relativas ao usuário.
5. Enquanto N <> Nulo
6.       Aplica Regras sobre N
7.       Se condições de acesso forem satisfeitas então
8.            N é adicionado a Visao
          Fim Se
           N ← Próximo Nó
   Fim enquanto
9 . Retorna Visao
```

Figura 4 - Algoritmo para leitura em Documentos

Na figura 4, o algoritmo de leitura monta uma lista de todos os nós de um documento a que o usuário tem acesso. A partir dessa lista é possível criar uma visão exclusiva de um documento para um usuário.

```
1. Recebe (Sujeito, Objeto, Conteúdo_Inclusão, Documento Alvo)
2. Verifica Grupo (Sujeito)
3. Verifica Politicas (Grupo, Sujeito)
   {O Conteúdo_Inclusão  representa o que se quer incluir}
4. E ← Conteúdo_Inclusão
   {A letra N representa um Nó na árvore de do documento XML}
   N ← Objeto
5. Aplica Regras sobre N
   Se condições de acesso forem satisfeitas então
6.    Escreve Elemento (E,N) {Esta função Escreve o Conteúdo E no
      Objeto N}
   Fim Se
```

Figura 5 - Algoritmo para escrita em documentos

No caso da escrita em documentos (figura 5) o algoritmo permite que possam ser modificadas ou criadas apenas as partes de um documento que o usuário tem acesso.

4. Resultados de Implementação

Para a implementação e validação do controle de acesso proposto são utilizados pacotes de desenvolvimento do conjunto de aplicações disponíveis para a linguagem Java. O pacote de desenvolvimento JDOM[5] possibilita a manipulação de dados em documentos XML de maneira fácil e eficiente, e implementa as funções de acesso tanto da API do SAX como as do DOM. Também é utilizado o pacote JAXEN[6] para navegação em documentos XML utilizando a linguagem XPath.

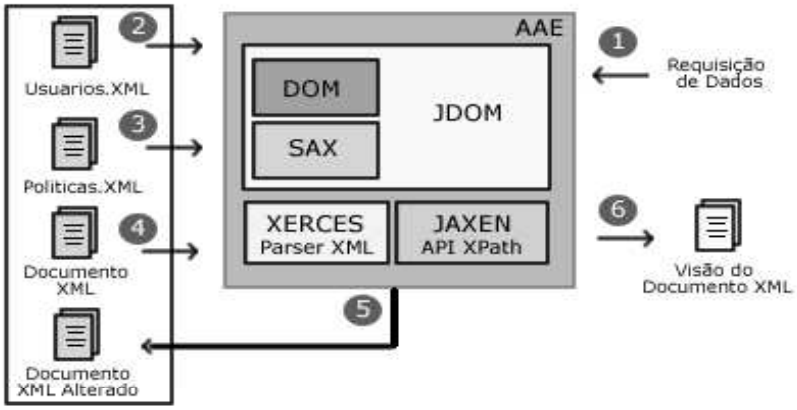

Figura 6 – Modelo para controle de leitura

A figura 6 demonstra em passos como o modelo interage quando uma aplicação *web* requisita uma leitura ou escrita de um documento XML, para um usuário qualquer. Uma aplicação *web* envia uma solicitação de leitura ou escrita a um documento XML para a *Aplicação de Autorização e Execução* (*AAE*), no passo 1. A *AAE* faz a autenticação do usuário através do documento de usuários no passo 2. O terceiro passo ocorre quando a *AAE* verifica as autorizações para o usuário baseado nas regras definidas para ele ou grupo para o qual o usuário pertence. As regras estão contidas no documento de políticas. No quarto passo, depois de verificada a procedência da requisição, a *AAE* processa a requisição sobre o documento XML, baseada no algoritmo de escrita ou leitura. No caso de escrita a *AAE* atualiza o documento XML e fornece ao usuário da aplicação *web* uma visão do resultado final da operação, passos 5 e 6 respectivamente. Se for uma leitura a *AAE* se encarrega de devolver apenas a visão dos dados requeridos para a aplicação *web*, passo 6.

A aplicação *AAE* implementa internamente os algoritmos de leitura e escrita apresentados anteriormente na seção 3. Ela faz uso das funções presentes nas APIs implementadas no JDOM juntamente com o *parser* XML Xerces[7] e o pacote JAXEN para realizar as operações de controle de acesso e manipulação de dados em documentos XML.

Na proposta de [Kudo 2000] são definidos dois módulos diferentes, chamados *Provisional Autorization Module* (PAM) e *Request Execution Module* (REM), utilizados para a autorização e execução de requisições, respectivamente. Diferente de [Kudo 2000] o modelo proposto utiliza a aplicação *AAE* como única responsável pela autorização e execução de requisições enviadas por uma aplicação *web*.

[5] http://www.jdom.org/
[6] http://jaxen.sourceforge.net/
[7] http://xml.apache.org/

5. Considerações Finais

Este artigo definiu um modelo de controle de acesso para conteúdos digitais, explorando as características e facilidades do XML como estruturação e semântica. O modelo permite que aplicações *web* possam manipular o conteúdo de documentos XML de uma maneira segura e eficaz, de acordo com os testes realizados.

Comparado com outros modelos, o modelo proposto apresenta algumas vantagens. A principal vantagem está na definição do algoritmo para escrita e alteração em documentos XML não presente nas propostas de [Damiani 2002] e [Gabillon 2001]. Já o algoritmo de leitura difere do algoritmo proposto em [Gabillon 2001], pois o modelo proposto na seção 3 mostra que é possível definir a participação não somente de usuários, mas também de seus respectivos grupos para a seleção das políticas envolvidas na operação. As semânticas para as regras de autorização são mais completas que as propostas em [Gabillon 2001] e [Damiani 2002] conforme os aspectos apresentados na seção 3 deste artigo.

Para trabalhos futuros pode ser feito uso da linguagem XACML[8] para definição de regras e especificação das políticas de segurança. Esse modelo está aberto para uso de outros padrões de segurança existentes que ainda não foram especificados.

Referências Bibliográficas

A. Gabillon, E. Bruno, “Regulating Access to XML documents”, Fifteenth Annual IFIP WG 11.3 Working Conference on Database Security. Niagara on the Lake, Ontario, Canada July 15-18, 2001.

E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Securing XML Documents," In Proc. of the 2000 International Conference on Extending Database Technology (EDBT2000), Konstanz, Germany, March 27-31, 2000.

M. Kudo and S. Hada. "XML Document Security based on Provisional Authorisation". In Proceedings of the 7th ACM Conference on Computer and Communications Security. November, 2000, Athens Greece, p. 87-96.

J. Clark. "XSL Transformations (XSLT) Version 1.0". World Wide Web Consortium (W3C).http://www.w3c.org/TR/xslt (November 1999).

J. Clark et al.. "XML Path Language (XPath) Version 1.0". World Wide Web Consortium(W3C). http://www.w3c.org/TR/xpath (November 1999).

T. Bray et al. "Extensible Markup Language (XML) 1.0". World Wide Web Consortium (W3C). http://www.w3c.org/TR/REC-xml (October 2000).

E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati. “A fine-grained access control system for XML documents”. ACM Transactions on Information and System Security (TISSEC) May 2002, pp. 169-202, Volume 5, Issue 2.

[8] http://www.oasis-open.org/committees/xacml