MAC5701 – Tópicos em Ciência da Computação
Monografia

Stefan Neusatz Guilhen
sneusatz@ime.usp.br

Controle de acesso baseado em papéis e certificados de atributos X.509

Orientador: Prof. Dr. Francisco Carlos da Rocha Reverbel
Departamento de Ciência da Computação

Universidade de São Paulo
- São Paulo, SP – 2005 -

1 Introdução

A grande maioria dos sistemas desenvolvidos atualmente tem como requisito algum tipo de controle de acesso às informações a aos recursos que são de grande importância. Primeiro, usuários devem se identificar ao sistema utilizando algum mecanismo de autenticação. Depois de identificado, o usuário deve somente poder acessar recursos para os quais ele tem permissão de acesso.

O RBAC [1] ou *Role-Based Access Control* é um mecanismo para controle de acesso baseado em papéis. Ele define um conjunto de papéis, que geralmente representam posições profissionais como gerente, administrador, diretor, etc, e atribui a cada papel um conjunto de permissões ou privilégios, que representam as ações que eles podem executar. Esse mecanismo simplifica de maneira significativa o gerenciamento de controle de acesso para um grande número de usuários, uma vez que tipicamente o número de papéis que os usuários podem desempenhar é muito inferior ao número de usuários em si.

Diversos serviços de segurança e *middlewares,* como servidores de aplicação, utilizam o RBAC como mecanismo de autorização e controle de acesso. Entretanto, a maioria dessas aplicações não utiliza uma maneira padronizada para mapear os usuários de uma aplicação aos seus respectivos papéis.

Recentemente, o comitê X9 da U.S. American National Standards Institute (ANSI) desenvolveu uma infra-estrutura de gerenciamento de privilégios, ou PMI [2],[3], que utiliza os certificados de atributos X.509 para armazenar o conjunto de privilégios de um usuário. Os certificados de atributos X.509 podem armazenar o conjunto de papéis de um usuário, estabelecendo dessa forma uma estrutura padronizada para fazer o mapeamento entre os usuários e seus papéis.

Essa monografia está organizada da seguinte forma: a seção 2 apresenta os modelos de controle de acesso definidos pelo RBAC. A seção 3 apresenta uma visão geral das infra-estruturas de chave pública (PKI) e de gerenciamento de privilégios (PMI), dos certificados de atributo X.509 e da relação entre a PMI e a PKI. A seção 4 discute os modelos “push” e “pull” de distribuição de credenciais (privilégios) utilizando os certificados de atributo X.509. Por fim, a seção 5 apresenta dois projetos estudados que fornecem um serviço de controle de acesso utilizando o RBAC e certificados digitais.

2 Role-Based Access Control - RBAC

Role-Based Access Control ou RBAC é um mecanismo de autorização (controle de acesso) que define um conjunto de papéis (roles) e atribui a cada papel um conjunto de permissões. Cada usuário é então associado a um ou mais papéis e herda todas as permissões desses papéis.

A grande vantagem associada ao uso do RBAC é a simplificação do processo de gerenciamento de permissões. Uma empresa pode ter facilmente milhares de

funcionários que são usuários de um sistema. Entretanto, pode-se observar que muitos usuários possuem exatamente os mesmos privilégios, pois desempenham o mesmo papel (ou função) dentro da empresa. A idéia é então agrupar os usuários em papéis, que representam sua função junto ao sistema e atribuir aos papéis e não aos usuários individuais o seu conjunto de permissões.

Existem quatro modelos de RBAC. O modelo simples, ou $RBAC_0$ é o modelo descrito acima: define diversos papéis, que geralmente representam posições profissionais tais como secretária, diretora e gerente. O administrador do sistema atribui a cada papel um conjunto de permissões que representam as ações que os papéis podem executar nos recursos protegidos e depois atribui um ou mais papéis para cada usuário real. Ao acessar um recurso, o usuário apresenta os seus papéis e o serviço de autorização determina o conjunto de permissões desses papéis para decidir se o acesso deve ou não ser permitido.

O modelo hierárquico, ou $RBAC_1$, é uma extensão mais sofisticada do modelo básico, permitindo que um papel estenda outros papéis e herde o seu conjunto de permissões. Assim, por exemplo, um papel gerente pode estender o papel funcionário para indicar que todos os privilégios alocados a um funcionário também se aplicam a um gerente, mesmo que isso não esteja explicitamente descrito. Esse modelo facilita ainda mais o gerenciamento de permissões, pois permite que permissões comuns a vários papéis sejam agrupadas em novo papel do qual os demais herdam, evitando replicação das permissões.

O modelo restrito, ou $RBAC_2$, é uma outra extensão do modelo básico. Embora a numeração possa sugerir que ela é uma extensão do $RBAC_1$, ela na verdade é uma outra extensão do $RBAC_0$. Com o $RBAC_2$, o administrador do sistema pode definir um conjunto de restrições para as permissões alocadas aos usuários. Uma restrição comum é declarar que certos papéis são mutuamente exclusivos, ou seja, a mesma pessoa não pode possuir os dois papéis ao mesmo tempo. Outras restrições podem restringir o número de papéis que um usuário pode ter ou então o número de pessoas que podem ter um dado papel. Já o modelo consolidado, ou $RBAC_3$, é uma extensão que inclui os modelos $RBAC_1$ e $RBAC_2$ ao mesmo tempo.

É importante salientar que o RBAC apenas define formalmente os modelos descritos acima. Ou seja, o RBAC não é uma framework para controle de acesso, apenas uma especificação. Implementações do RBAC são livres para definir como a política de controle de acesso a um recurso deve ser descrita, como papéis devem ser descritos e como associar os papéis aos usuários. Na prática, observa-se o uso de XML como linguagem para definir a política de acesso e também os papéis e suas relações/restrições.

3 PKI e PMI

A infra-estrutura de gerenciamento de privilégios, ou simplesmente PMI, surgiu a partir da necessidade de um mecanismo forte de autorização e que fosse independente do mecanismo de autenticação. O padrão X.509, juntamente com a

ITU-T e a ISO/IEC [2] havia definido uma infra-estrutura de chave pública, a PKI, cujo elemento central é o certificado de chave pública, também conhecido por public key certificate ou PKC. O principal foco dessa infra-estrutura era prover um mecanismo forte de autenticação.

Os PKCs são documentos digitalmente assinados por uma entidade certificadora – a Certification Authority ou CA – e associam a identidade de um usuário a uma chave pública. Na prática, o uso da PKI revelou a necessidade de se armazenar outros tipos de dados em um certificado, além da chave pública e da identidade do seu portador. Por isso, versões recentes do padrão X.509 definem uma série de extensões no PKC para o armazenamento de informações como, por exemplo, dos papéis que um usuário desempenha, seus privilégios ou outro tipo de informação de autorização. Porém utilizar as extensões dos PKCs para armazenar informações a respeito de autorização gerou alguns efeitos negativos, dentre os quais os que mais se destacam são:

- Em primeiro lugar, as informações de autorização tipicamente não têm o mesmo tempo de validade da identidade e da chave pública que são armazenadas em um PKC. Em geral, informações de autorização tendem a ter validade mais curta do que a identidade e chave pública. Por isso, se essas informações são colocadas na extensão de um PKC, o tempo durante o qual o PKC deveria ser válido é encurtado, pois modificações das informações de autorização irão requerer que um novo PKC seja emitido e o antigo seja revogado.

- Em segundo lugar, dificilmente a entidade que emite um certificado de chave pública terá autoridade para estabelecer informações de autorização e controle de acesso. Como resultado, uma entidade emissora de PKCs precisa realizar algumas operações a mais para obter essas informações de uma ou mais fontes de autorização.

Reconhecendo que os certificados de chave pública (PKCs) não são a melhor estrutura para carregar informações de autorização, o comitê X9 da U.S. American National Standards Institute (ANSI) desenvolveu umas abordagens alternativas, conhecidas como certificados de atributos ou ACs. Semelhante ao certificado de chave pública, que associa uma chave pública a uma identidade, um AC associa um conjunto de atributos ao seu portador. A edição 4 do padrão X.509 é a primeira a totalmente padronizar um mecanismo forte de autorização, que foi chamado de infra-estrutura de gerenciamento de privilégios ou PMI. A estrutura de dados fundamental dessa infra-estrutura é o X.509 Attribute Certificate, ou X.509 AC.

A entidade emissora dos certificados de atributos é chamada de Attribute Authority ou AA. Os X.509 ACs são digitalmente assinados pela AA. Quando as permissões de autorização de um usuário são revogadas, a Attribute Authority emite uma lista de certificados de atributos revogados (Attribute Certificate Revocation List ou ACRL) contendo a lista de ACs que não devem mais ser aceitos.

Os certificados de atributos estabelecem uma separação clara entre o processo de autenticação (identificação) e autorização (controle de acesso). Com eles, entidade emissora dos PKCs – Certification Authority – não tem mais a responsabilidade de obter de alguma forma os atributos de autorização de um usuário. Ela precisa apenas se preocupar com os atributos relacionados à identidade do mesmo. A Attribute Authority assume a responsabilidade pelos atributos de autorização de um usuário.

Assim como a Certification Authority não se preocupa com informações de autorização, a Attribute Authority não se preocupa com informações sobre a identidade de uma pessoa, apenas com inferências que podem ser feitas a seu respeito uma vez que sua identidade foi provada. Isso traz as seguintes vantagens:

- Promove a interoperabilidade na medida em que favorece o gerenciamento distribuído de privilégios e atributos de autorização. Como esses atributos de autorização não são emitidos pelas Certification Authorities, é possível distribuir o gerenciamento desses atributos por várias Attribute Authorities, cada uma fornecendo informações de autorização em um contexto diferente. Por exemplo, em um país é possível que cada cidadão tenha um PKC assinado por uma única Certification Authority e que comprova sua identidade. Os seus privilégios nos mais diversos contextos da vida social podem ser emitidos por diferentes Attribute Authorities. Uma AA pode emitir CAs contendo informações sobre os papéis do cidadão no sistema eleitoral (eleitor, mesário, etc) enquanto outra pode emitir CAs contendo os atributos de sua carteira de motorista, identificando a sua autorização para dirigir determinados tipos de veículos.

- Separação de jurisdição. Como os certificados de atributos são emitidos por autoridades que realmente possuem as informações de autorização (AAs), as Certification Authorities não precisam mais coletar essas informações e isso evita delegação de responsabilidades para as CAs.

- Certificados de atributos podem ter um tempo de vida muito mais curto do que os certificados de chave pública e podem ser revogados separadamente. Isso é uma conseqüência imediata da remoção dos atributos de autorização dos certificados de chave pública. Se os atributos de um usuário mudam apenas os seus respectivos ACs precisam ser revogados sem causar nenhuma revogação de seu PKC.

A figura abaixo ilustra a definição de um certificado de atributos de acordo com a RFC3281 [3]:

```
AttributeCertificate ::= SEQUENCE {
     acinfo               AttributeCertificateInfo,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING
}
```

```
AttributeCertificateInfo ::= SEQUENCE {
     version              AttCertVersion -- version is v2,
     holder               Holder,
     issuer               AttCertIssuer,
     signature            AlgorithmIdentifier,
     serialNumber         CertificateSerialNumber,
     attrCertValidityPeriod   AttCertValidityPeriod,
     attributes           SEQUENCE OF Attribute,
     issuerUniqueID       UniqueIdentifier OPTIONAL,
     extensions           Extensions OPTIONAL
}
```

Figura 1 – Certificado de Atributo

4 - Distribuição de Credenciais – modelos “push” e “pull”

Existem dois modelos fundamentais para distribuição de certificados contendo as credenciais (permissões, papéis, etc) de um usuário: o modelo “push” e o modelo “pull”. O modelo “push” é utilizado em ambientes onde é necessário que o cliente empurre (push) seus certificados de atributos para o servidor. O resultado disso é que o servidor não precisa fazer nenhum tipo de busca para encontrar os atributos do cliente, o que melhora o desempenho. O modelo “push” é ideal nos casos em que os direitos e privilégios do cliente são definidos e atribuídos no domínio do cliente.

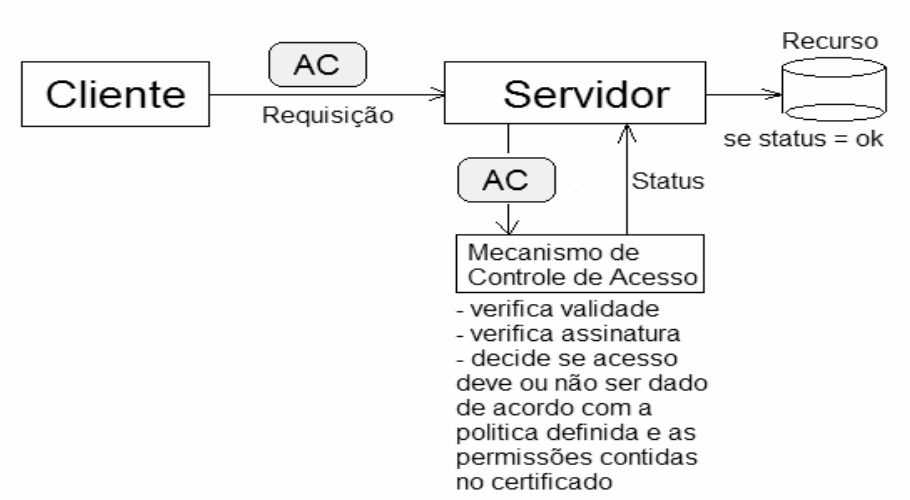

Figura 2 – Modelo Push de propagação

No modelo “pull”,o cliente simplesmente se autentica e o servidor por sua vez “puxa” (pull) os certificados de atributos do cliente de algum repositório. Ele é ideal nos casos em que os privilégios do cliente são definidos e atribuídos no domínio do servidor.

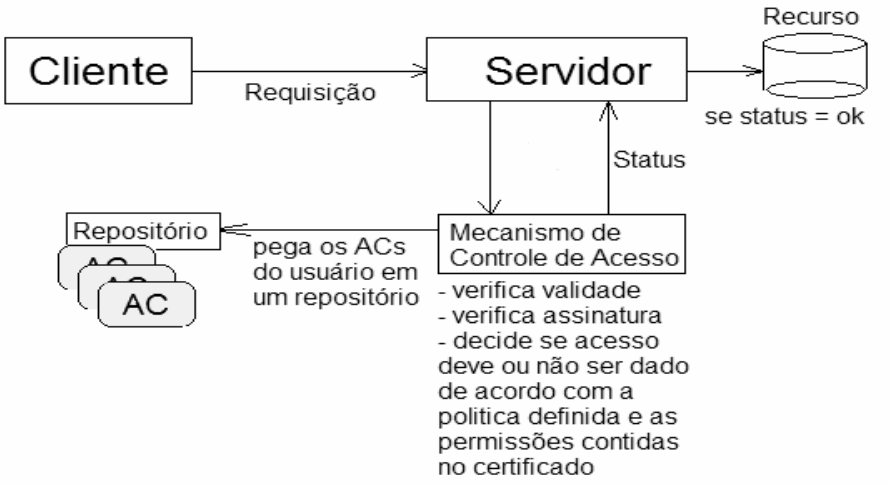

Figura 3 – Modelo Pull de propagação

Tipicamente o modelo "push" é aplicado da seguinte forma: uma requisição de acesso é submetida ao serviço de autorização juntamente com o certificado de atributos do cliente que originou a requisição. O serviço de autorização decide se o acesso deve ou não ser permitido baseado nas informações contidas nos atributos do certificado, na política de controle de acesso em vigor e na validade do certificado.

Vários fatores podem colaborar para que um certificado seja considerado inválido, mas os dois principais são: o certificado expirou (ou seja, seu prazo de validade venceu) ou ele foi revogado, seja porque o portador não é mais considerado um usuário aprovado do sistema ou porque seu conjunto de privilégios foi alterado. O primeiro caso é fácil de se verificar já que o certificado contém o prazo de validade em um dos seus atributos. Já a revogação é mais complicada de se tratar no modelo push.

A solução usual adotada é de fazer com que a autoridade emissora do certificado publique listas de revogação de certificados (CRLs) contendo os certificados que não devem mais ser considerados válidos. O principal problema com as CRLs é que essas listas são publicadas periodicamente, de forma que a revogação de privilégios de um certo usuário não entre em vigor imediatamente. Outros mecanismos também foram propostos, como, por exemplo, o OCSP [5], que é um protocolo que permite a verificação online da validade de um certificado junto à entidade certificadora.

Já no modelo pull, o serviço de autorização é responsável por obter as informações necessárias ao invés de depender dos dados fornecidos pelo usuário. Nesse modelo, as informações de autenticação são mantidas no lado servidor sob a forma de certificados de atributos. Esses certificados são em geral armazenados em repositórios confiáveis – tipicamente serviços de diretório LDAP. A revogação de certificados é suportada diretamente por esse modelo através da simples

remoção do certificado de atributo correspondente. Uma discussão mais detalhada sobre o assunto pode ser encontrada em [4].

5 Projetos estudados

Akenti Authorisation Infraestructure

O Akenti é uma infra-estrutura de autorização desenvolvida no Lawrence Berkeley National Laboratory [7]. O principal objetivo do projeto é de fornecer uma maneira de expressar e fazer valer uma política de controle de acesso de forma distribuída, evitando os problemas que aparecem quando uma única pessoa é responsável por gerenciar e garantir os requisitos de controle de acesso.

O controle de acesso aos recursos tem sido feito por um longo tempo com o uso de listas de controle de acesso. Essas listas são gerenciadas e controladas por uma única pessoa, que é responsável pela manutenção da política de controle adotada. O problema é que nem sempre é adequado centralizar esse tipo de gerenciamento, pois é possível que mais de uma pessoa tenha autoridade para decidir como o acesso ao recurso deve ser feito. Nesse caso, o uso de listas de controle de acesso é totalmente inadequado, pois o caráter centralizado desse tipo de mecanismo impede que mudanças feitas por uma das pessoas envolvidas possam ser aplicadas imediatamente. Uma requisição tem que ser feita para o controle central, que tem que verificar se a requisição foi originada por uma parte autorizada, checar se a requisição não foi alterada durante o envio e só então aplicar as mudanças apropriadas. Isso compromete a escalabilidade, principalmente nos casos em que as partes envolvidas na definição da política de controle de acesso estão distribuídas geograficamente.

O Akenti foi desenvolvido para lidar com essa situação, promovendo o gerenciamento distribuído da política de acesso aos recursos. Nele, uma parte envolvida na definição da política pode gerenciar os seus requisitos de controle de acesso independentemente de outras partes. Além disso, cada parte pode modificar seus requisitos a qualquer momento e pode ter certeza de que as mudanças irão entrar em vigor imediatamente. O Akenti realiza decisões de autorização baseado em um conjunto de documentos digitalmente assinados que possuem as instruções de autorização. A PKI e protocolos seguros de troca de mensagens fornecem autenticação de identidades e integridade de mensagens respectivamente.

A figura a seguir ilustra a arquitetura e os principais componentes do Akenti:

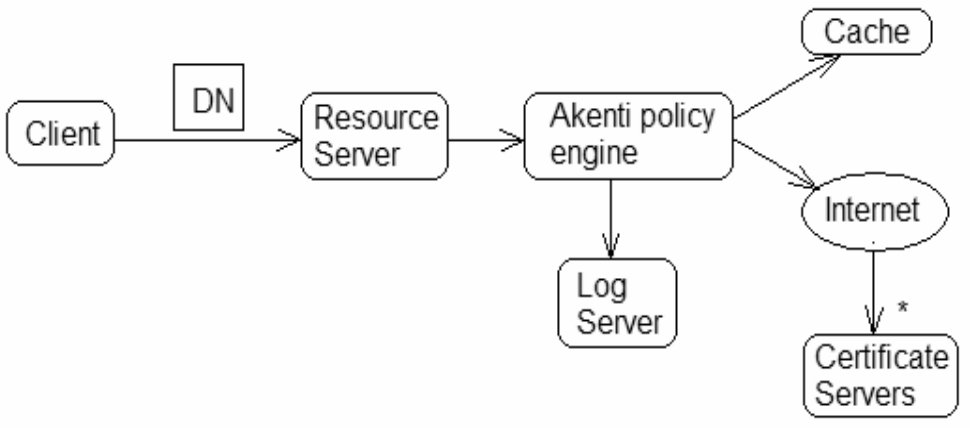

Figura 4 – Akenti - Arquitetura

No Akenti, os recursos a serem protegidos são configurados como Resource Servers. Clientes se autenticam aos Resource Servers utilizando certificados de chave pública X.509. As requisições são delegadas ao Policy Engine, que é responsável por encontrar as políticas de acesso definidas para o recurso, as credenciais do cliente e decidir se o acesso deve ou não ser concedido. Tanto as políticas de acesso quanto as credenciais do cliente são armazenadas em certificados digitais que se encontram distribuídos em diversos Certificate Servers. Todas as ações do Policy Engine são registradas no Log Server para auditoria. Um cachê pode ser opcionalmente utilizado para evitar que o Policy Engine tenha que procurar sempre pelas credenciais de um cliente que já fez requisições anteriormente.

Para implementar o gerenciamento distribuído de políticas de controle de acesso, o Akenti faz uso de uma série de certificados digitais. As pessoas que possuem autoridade para criar políticas de acesso geram certificados de condição de uso, que são criados e assinados pelo Use-Condition Generator. As Attribute Authorities nas quais o Policy Engine confia podem gerar e assinar certificados de atributos contendo as credenciais dos usuários através do Attribute Generator. Já as Certification Authorities confiáveis geram os X.509 PKCs que são usados pelo cliente durante a autenticação. A figura a seguir ilustra a geração dos certificados:

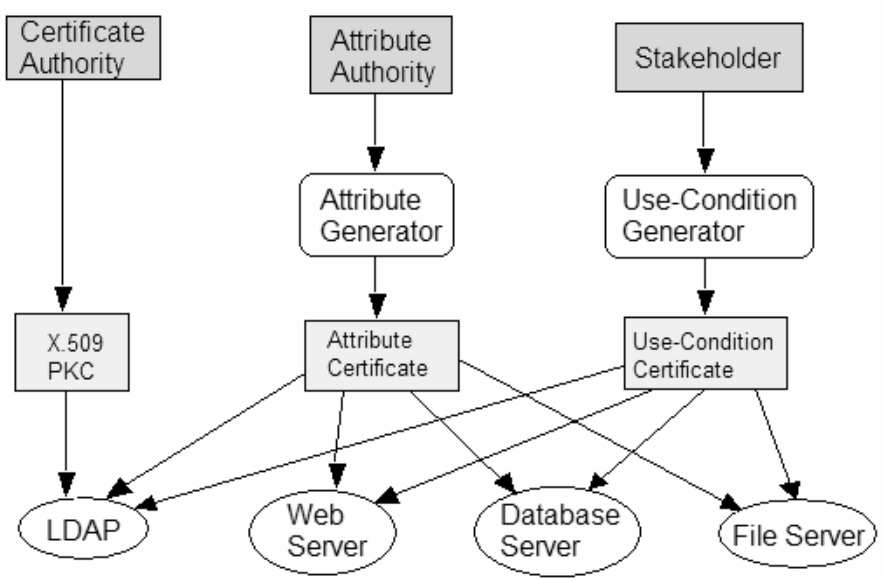

Figura 5 – Akenti - Certificados

O Akenti suporta o RBAC se a política de condição de uso for descrita em função dos papéis dos usuários e os certificados de atributos gerados pelas AAs contiverem os papéis de cada usuário como credenciais. Entretanto o Akenti é bem flexível nesse ponto e permite que outros métodos de controle de acesso sejam aplicados, não apenas o RBAC. Como desvantagens, podemos citar que a única forma de autenticação suportada pelo Akenti é através de certificados de chave pública, dificultando sua utilização em recursos que aceitam outras formas de autenticação, como login-senha ou análise impressão digital, por exemplo. Além disso, todos os certificados gerados pelo Akenti são proprietários, incluindo os certificados de atributos, que não aderem ao padrão proposto pela PMI que usa os X.509 ACs como certificados de atributos.

PERMIS

O PERMIS [6] (PrivilEge and Role Management Infraestructure Standards) é uma infra-estrutura de autorização financiada pela European Commission (EC) e desenvolvida no instituto de segurança da informação da universidade de Salford, UK. O objetivo do projeto é implementar uma PMI baseada nos certificados de atributos X.509.

O Permis utiliza o RBAC como mecanismo de controle de acesso. Todos os dados necessários para decisões de autorização, como a especificação dos papéis e os privilégios alocados a cada papel, são descritos por uma política de autorização, que é por sua vez armazenada em um certificado digital para garantir sua integridade. O Permis escolheu o XML como linguagem para descrição dessa política porque XML possui um extenso conjunto de ferramentas de suporte e tem se consolidado como um padrão na indústria. Algum tempo depois do Permis ter definido o seu DTD, o consórcio Oasis começou um trabalho para definir a

Extensible Access Control Markup Language ou XACML [8]. Apesar do Permis não aderir à XACML, muitas similaridades existem entre os dois formatos de descrição da política de controle de acesso.

Tanto o AC contendo a política de acesso bem como os ACs contendo os papéis associados a cada usuário são armazenados em serviços de diretório LDAP distribuídos, implementando um modelo "pull" para distribuição dos ACs. Nesse caso, as listas de revogação (CRLs) não são necessárias uma vez que a entidade emissora dos certificados (Attribute Authority) pode simplesmente remover os certificados que não são mais válidos do seu repositório.

A arquitetura do Permis é composta por um subsistema de alocação de privilégios, que é responsável por alocar privilégios aos usuários, e um subsistema de verificação de privilégios, que é responsável por autenticar e autorizar usuários. O componente principal do subsistema de alocação de privilégios é Privilege Allocator. Através dele, as Attribute Authorities podem associar papéis aos usuários na forma de certificados de atributos. É ele quem gera e assina os certificados de atributos contendo os papéis dos usuários e também o AC que contém a política de controle de acesso. Os ACs gerados são armazenados em um diretório LDAP.

Já o subsistema de verificação de privilégios autentica e autoriza usuários, verificando suas permissões de acesso. A funcionalidade desse subsistema foi dividida em duas partes: um componente específico de aplicação para realização da autenticação, chamado de AEF ou Access-Control Enforcement Function, e um componente independente de aplicação, chamado de ADF ou Access-Control Decision Function. Cada aplicação pode definir a sua forma de autenticação independentemente da autorização (por exemplo, utilizando PKI com X.509 PKCs ou simplesmente um par usuário-senha). A comunicação entre a AEF e a ADF é feita através da Permis API, uma API baseada na AZN API [9] definida pelo Open Group. A figura a seguir ilustra a arquitetura do Permis, com seus principais componentes:

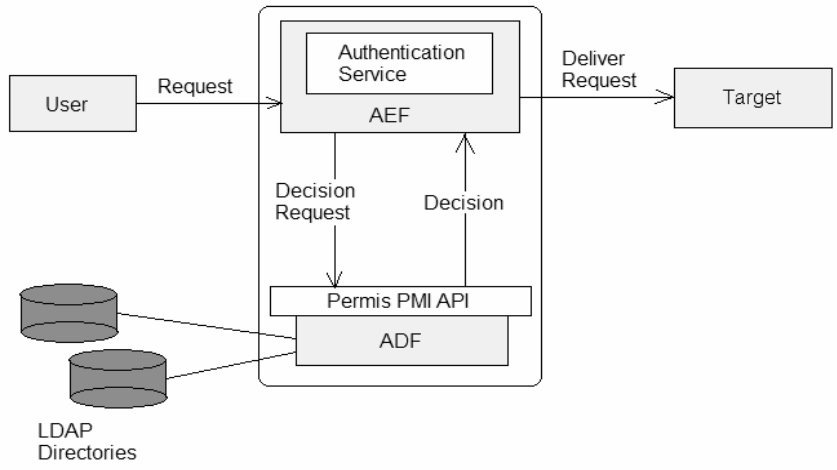

Figura 6 – Permis – Arquitetura.

Como podemos observar, o Permis fornece um mecanismo de autorização para proteção de recursos baseado no RBAC, implementando inclusive o modelo $RBAC_1$. Ao contrário do Akenti, o Permis é bem flexível quanto ao mecanismo de autenticação utilizado e suporta qualquer forma de autenticação. Além disso, ele adere aos certificados de atributo X.509 definidos pela PMI, o que facilita sua interoperabilidade com diversas Attribute Authorities. No Akenti, as AAs envolvidas precisam gerar os certificados usando ferramentas próprias do Akenti, pois o formato do certificado não é padronizado.

6 Conclusão

Uma grande parte dos sistemas que foram desenvolvidos nos últimos anos utiliza algum modelo de RBAC para suportar controle de acesso. Esse mecanismo está presente desde frameworks específicas de segurança até servidores de aplicação, o que comprova o seu fortalecimento e amadurecimento como meio para fornecer controle de acesso a recursos computacionais. O Permis é um exemplo de sucesso do uso do RBAC em combinação com os certificados de atributos X.509 definidos pela PMI para prover um serviço de autorização robusto e confiável.

7 Referências

[1] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, *Role based access control models*, IEEE Comput. 29, 1996.

[2] ITU-T Rec. X.509 ISSO/IEC 9595-8, *The Directory: Public-key and Attribute Certificate Frameworks*, 2001.

[3] *An Internet Attribute Certificate Profile for Authorization*, Abril 2002, http://www.ietf.org/rfc/rfc3281.txt

[4] J. Crampton, H. Khambhammetu, *Authorization and Certificates: Are we pushing when we should be pulling?* Proceedings of the IASTED International Conference on Communication, Network, and Information Security, 2003, p 62-66

[5] *Online Certificate Status Protocol – OCSP*, Junho 1999, http://www.ietf.org/rfc/rfc2560.txt

[6] D.W. Chadwick, A. Otenko, *The PERMIS X.509 role based privilege management infrastructure*, Future Generation Computer Systems, Volume 19, Edição 2, Fevereiro 2003, p 277-289.

[7] *Akenti Authorization Infrastructure*. http://dsd.lbl.gov/Akenti

[8] OASIS eXtensible Access Control Markup Language http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[9] Open Group. “Authorization (AZN) API”, Janeiro 2000, ISBN 1-85912-266-3