Privilege Management Infrastructure

Darren P. Mundy
University of Hull, UK

Oleksandr Otenko
University of Kent, UK

PRIVILEGE MANAGEMENT INFRASTRUCTURE: AN OVERVIEW

Public Key Infrastructures (PKI) are now in place in a number of organizations, and there is a wide amount of material available that can be used to obtain familiarisation with the concept (Adams & Lloyd, 2002; Housley & Polk, 2001; Nash, Brink, & Duane, 2001). Although related to PKI, Privilege Management Infrastructure (PMI) is a recent development in the network security field. PMI has been designed to supply the authorization function lacking in the PKI model. This article will provide an overview of PMI, state the relationship between PKI and PMI, and will finally provide a number of examples of present PMI architectures such as PERMIS (Chadwick & Otenko, 2002), OASIS Active Security (Yao, Moody, & Bacon, 2001) and AKENTI (Thompson, Essari, & Mudumbai, 2003).

WHAT IS PMI?

PMI can generally be thought of as the infrastructure supporting a strong authorization subsystem via the management and use of privileges (Adams & Lloyd, 2002). PMI is essentially a term used to encompass the management of authorization processes such as access control, rights management, levels of authority, delegation of authority, and so on. A PMI helps an organization to provide secure access to any target resource they specify based on policy. A policy should detail such information as which users are allowed access to which resources, what actions they are allowed to perform, when they are allowed access, for example, time constraints, what privileges they need to be able to access the resource and carry out an operation.

As organizations embrace electronic business they are increasingly striving to provide electronic access to greater amounts of organizational resources to improve their services and decrease transaction costs. However, by opening up electronic access to their resources for their partners, clients, employees, and so on … they are heightening the security risks that they face (Newman, 2003). Organizations need to be sure that access to their resources is controlled by a variety of security mechanisms, for example:

1. To ensure the party requesting access is who they say they are (authentication).
2. That the party has sufficient rights to access the resource (authorization).
3. That confidential material is only read by those authenticated and authorized parties (privacy).
4. That the transaction is monitored (audit and control).

PMI addresses only authorization. To address other points, corresponding subsystems should be deployed. In further sections it will be explained how the PMIs and PKIs are related, and examples of the use of PMIs will be provided. The discrepancies between various implementations will be highlighted, and the difficulties with using PMIs will be discussed.

The Relationship Between PKI And PMI

The authorization subsystem supported by a PMI can be relied upon to control access based on the privileges possessed by a user. However, it doesn't provide any assurance as to the user's identity. To ensure identity we require an identity management system such as a PKI. Familiarity with the PKI concept can be obtained using the wealth of available literature

(Adams & Lloyd, 2002; Housley & Polk, 2001; Nash et al., 2001). In this context however it can simply be stated that a PKI provides authentication services whilst the PMI provides authorization services. There is a large similarity between PKI and PMI at multiple levels (Chadwick & Otenko, 2002), for example.

- In a PKI users are given digital certificates proving their identity; in a PMI users are given digital certificates proving their privilege(s).
- In a PKI a Certificate Authority issues the digital certificates; in a PMI an Attribute Authority creates the digital certificates.

PMI ARCHITECTURES FOR TRUST ESTABLISHMENT

In this section, Privilege Management will be looked at from the point of view of the access control system.

Prior to the introduction of Privilege Management Infrastructures (PMI), access control systems trust only their "local" information about the outer world. This is very effective for small groups of people (e.g., multi-user Operating Systems). However, when the number of users willing to co-operate increases, it becomes more difficult to reflect all of the circumstances of the world locally. Dynamicity of relationships between the resource owner and the users accessing the resource also increases the difficulty of managing the privileges each of the users has, limiting scalability of such systems.

To facilitate scalable solutions, trust in the people must be established in a distributed manner, and a means of distributing trust is required. This can be achieved in a number of ways. This section describes how this is done in three different PMI models. It starts with the approach adopted by X.509 and is followed by description of the Akenti and Active Security PMI architectures.

X.509

In X.509 there is a single root of trust, the Source of Authority (SOA). It stands for the owner of a resource, or an agent acting on his behalf. The SOA specifies the rules for establishing trust relationships, and access control rules. All such rules are written in a form of a policy, which governs the access control system. The SOA is also the ultimate authority in assigning privileges to end-entities, which will use the resource.

The SOA distributes the privilege to assign privileges to other entities, which are called Attribute Authorities (AAs), and the process of assigning this privilege is called delegation. These authorities, in their turn, may be allowed to assign privileges to end-entities, or delegate them further to other Attribute Authorities. Thus the PMI forms a tree of authorities, with a singular root, which is the Source of Authority. The leaf nodes are end-users, who can only assert their privileges, and cannot delegate them to other entities.

The fact of assignment of privilege to an entity (either to AA or to an end-user) is noted as an X.509 Attribute Certificate (AC), which is a digitally-signed document, describing who has assigned what privilege to what entity. The privilege in such ACs is specified in a form of a privilege attribute that has to be interpreted by the access control system.

The access control system can discover trust relationships between the SOA (the resource owner) and the end-entities by obtaining their ACs and validating their contents using the policy written by the SOA. To achieve this, the access control system must obtain the ACs of the end-entity attempting access, the ACs of the Authority assigning the privilege to it (remember, that X.509 ACs specify who the grantor was), and ACs of all AAs that granted the privilege to do this to the authority, and to each of those AAs. Then the system needs to validate each of the assignments that occurred against the policy: if so, then the end-entity has been assigned a privilege in a trustworthy way, and an access control decision can be made; if the assignment of some privilege is not allowed by the policy, the privilege assignment is not trustworthy and should be discounted when making an access control decision.

In X.509, privilege assignment is valid if the granted privilege is a subset of all the privileges the grantor has, the only exception being the source of authority, which can assign any privilege to any entity[1]. To be able to make judgments if a granted set of privileges is a subset of the privileges the grantor had, the privilege attribute values must have order. Some access control models (MAC, RBAC) naturally have ordering of privilege attribute values; other models may need enhancement (Otenko, 2004).