How Role Based Access Control is implemented in SESAME

Mark Vandenwauwer, René Govaerts, Joos Vandewalle
Katholieke Universiteit Leuven
Dept. Elektrotechniek, ESAT-COSIC
Kardinaal Mercierlaan 94
B-3001 Heverlee - BELGIUM
E-mail : mark.vandenwauwer@esat.kuleuven.ac.be

Abstract

In this paper we want to share our experiences with implementing a scheme that enforces role based access control in a distributed, heterogeneous computing environment. This work was done in the framework of the EC-RACE project SESAME (A Secure European System in A Multi-vendor Environment). The SESAME project relies on the work done by ECMA (European Computer Manufacturers Association) to represent the credentials of the users and fully supports the GSS-API to help the application developers. We conclude that enforcing such a scheme is realistic and that writing applications, that benefit from the advantages of role based access control, is very feasible. We have built several demonstration applications.

1. Introduction

In recent years, many organizations have shifted their computing facilities from central mainframes (accessed from simple terminals via serial lines) to servers accessed from personal computers via a local area network (LAN). The switch to LANs removed some old problems and introduced some new ones. Many of these new challenges are not particularly related to security. For example, backup and recovery from equipment failures require a different approach in a client-server architecture. However, this paper will only consider security problems; moreover, only a limited number of them will be addressed.

If one does not know where data has come from or where it is going to, it is difficult to enforce any kind of security policy. It is therefore essential that one has an effective form of authentication that works in the new LAN environment. The authentication within SESAME is based upon the Kerberos [13] system but adds several extra features : logon using digital signatures, cross-realm authentication using public key cryptography [15, 21, 22]. Authentication on its own is however not sufficient. Access control and access control management facilities are also needed, and particular user communities may need other security services beyond that. This paper will address the access control issue.

Nowadays [19, 18] the trend towards role based access control (RBAC) has become apparent. In RBAC, access decisions are made based on the role of the user that is trying to run an application or access a resource. The main benefit of using RBAC is an improvement in the manageability of the system. Because administrators only have to maintain a list of who is entitled to exercise which role, it is for example easy for them to move a user from one department to another, or to add a user to the system. Access control at the application side is even more straightforward. The only thing one needs to do is to build an access control list (ACL) that specifies which roles are entitled to perform which actions. There is no need to care about who the user really is, only their role is important. This type of access control is also very user friendly. The user can log in with his rolename (e.g. secretary) instead of having to remember a username. They could even be granted a default rolename so that they can start working immediately upon entering the system. Experience has taught us that there is a good correlation between the needs in the real world and RBAC.

Here is a short outline of the rest of our paper. In the second paragraph we will describe the circumstances that lead to our work. In Section 3 the structure of SESAME will be detailed. We will talk about the different definitions of the term "identity" in Section 4. The Privilege Attribute Certificate (PAC) used to implement RBAC will be explained in Section 5. In Section 6, we will compare attribute certificates to the well known X.509 based certificates. An example of how to secure an Intranet will be given in Section 7. We will finish with our conclusion.
2. History

In the late 1980’s, confronted with a list of problems, the European Computers Manufacturers’ Association (ECMA) started work to develop standards for open systems. The work of ECMA was based on the standards formulated in the OSI Architecture [10], that describe the security services and their position in the seven layer OSI basic reference model, and the series of Security Frameworks developed in ISO/IEC JTC1 [12]. A Technical Report [4], published in 1988, concentrates on the application layer and describes a security framework in terms of application functions necessary to build secure open systems. The continuation of this report [5], defines the abstract security services for use in a distributed system. A parallel standard [6] describes a model for establishing secure relationships between applications in a distributed system. ECMA has now completed the work to define the functionality and the protocols for a distributed security service in charge of authenticating and distributing access rights to human and application principals, together with key distribution functions. The ECMA standard that is the result of this work is the ECMA-219 [7]. It was approved by the ECMA General Assembly in December 1994 and released in January 1995. The SESAME (Secure European System in A Multi-vendor Environment) project was an initiative of the European Commission to show that an implementation of these standards was both feasible and practical. In the last few years SESAME has evolved considerably. The latest release (V4) can be obtained from [20].

3. SESAME

In SESAME, access control can be managed centrally by a security administrator. There is a central database of the access control information associated with each user. This is what we call the Privilege Attribute Server (PAS). The access control information is transferred to applications using Attribute Certificates. SESAME implements the ECMA-219 Privilege Attribute Certificate (PAC) [7]. When a client wants to access a target application, he has to show that he owns a PAC permitting him to act in the necessary role. Mainly, there are two types of PAC:

1. Non-delegatable PACs. These are bound to a particular identity. They are protected by the Primary Principal Identifier (PPID) method [21, 23]. The security information inside the PAC gives more information about the particular session. Access control decisions may need to take into account the relative security of the client computer the user is logging in from.

2. Delegatable PACs, which act like capabilities. With these, it is possible to temporarily delegate some of a user’s access rights to another user/server. It remains important to keep this user/server accountable for its actions. Therefore, each entity in the system has its own identity, and is always authenticated as that identity. To enforce this, a delegatable PAC also contains a Check Value (CV). This CV is the result of applying a one-way function to a randomly chosen Protection Value (PV). When the user obtains its PAC, the PAS will also forward him his PV, encrypted under the current session key. Now when a user wants to forward his rights, he will send the PV (encrypted under a new session key) to a user/server. Now this user/server is able to prove that he knows the corresponding PV (to the CV inside the PAC). It is good practice to make the rights, which are conveyed, as restricted as possible.

Both types of PAC are issued with short expiry times (of the order of a few hours) to limit the time a compromised key or capability can be used for. Delegatable PACs can be used to achieve pseudo anonymity, provided that this is permitted by the security policy. A client that wishes to be anonymous obtains a delegatable PAC that does not contain its identity, and then delegates this PAC to a proxy server that acts on its behalf. The ultimate target of the access will know the identity of the proxy server but not the client behind it. The audit trails contain sufficient information to enable a security administrator to unmask an anonymous user if this should be required.
The PAC format is independent of the domain’s security policy. The details of the security policy are contained in the system components which create or interpret PACs: the PAS and each application server’s access control logic. The current SESAME implementation assumes a particular form of role-based policy: for a login session, each user takes on exactly one role; roles are enumerated and assigned identifiers; for each user, there is a list of the roles in which that user can act; and the access rights of a user are determined by the role in which they are acting. If a customer needs a different style of security policy then the privilege attribute server can be changed to enforce different rules or set(s) of rules.

4. Identities

In our implementation we assign different meanings to the word ‘identity’. Unlike the real world, in computer systems the term identity has multiple uses. Here are those that are of interest to the SESAME users:

1. Authenticated Identity: the name the user offers to log on to the system.
2. Access Identity: the name upon which applications make access decisions.
3. Audit Identity: the name under which one will be accountable.
4. Charging Identifier: if computer systems or other resources, that have to be paid for, are used then a billing address should be known.

5. The Privilege Attribute Certificate (PAC)

This section describes the structure and contents of the PAC, and how its contents are used.

5.1. PAC Structure

The general form of a security certificate has three components: (see Table 1)

- Information common to all security certificates, realized in the Common Contents and Signature fields,
- Security information specific to one or more security services (in this case access control), realized in the access privileges carried in the Specific Contents field,
- Information to control and/or limit the use of the security information, realized in the protection methods and audit identity carried in the Specific Contents field.

The SESAME PAC is based on a profile of the ECMA PAC. The ECMA PAC has been designed to conform to emerging ISO definitions of a security certificate, and in particular of an Access Control Certificate. A complete definition can be found in [7]. The individual fields of the SESAME PAC are described in the following paragraphs.

5.1.1. Common Contents

See Table 2.

<table>
<thead>
<tr>
<th>Table 2. Common contents of the PAC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Issuer Domain</strong></td>
</tr>
<tr>
<td><strong>Issuer Identity</strong></td>
</tr>
<tr>
<td><strong>Serial Number</strong></td>
</tr>
<tr>
<td><strong>Creation Time</strong></td>
</tr>
<tr>
<td><strong>Validity</strong></td>
</tr>
<tr>
<td><strong>Algorithm Identifier</strong></td>
</tr>
</tbody>
</table>

5.1.2. Specific Contents

Protection Methods

Fields used to control the use of the PAC (PPID, PV/CV, ...).

Privileges

The privileges granted by the PAC. All privilege attributes are associated with a defining authority, either explicitly by prefixing it with a defining authority field or by default. In SESAME, only attributes with a default defining
authority are created. The default is defined to be the value in the Issuer Domain field of the PAC. This means that privilege attributes contained in the PAC are always from the security domain of the PAS. However for ease of future extension, if an explicit defining authority is detected for an attribute by a PVF (the PAC Validation Facility is a trusted piece of code that checks the validity of the PAC at the target side of the communication) then it will be accepted under the proviso that it matches the Issuer Domain field in the PAC. Otherwise it is ignored (although the PAC is not rejected). In a later implementation the PVF check can be made more elaborate, by relating it to the defining authorities for which an issuing PAS is trusted. The check is necessary at this stage in order to protect against a compromised PAS. In SESAME, privileges can have any of a number of syntax representations. The attributes are:

- **Access Identity**: If a specific Access Identity for a user has been specified in the PAS database then this is used. Otherwise, the value defaults to the Authenticated Identity supplied in the PAC request. It is used for access control by the target computer’s access control logic.

- **Role Attribute**: This may be the default role attribute. It is an attribute corresponding to the role-name specified in the PAC request. It is used for access control decisions.

- **Primary Group**: The main access group of which the PAC owner is a member.

- **Secondary Group**: The secondary access group(s) of which the owner of the PAC is a member.

The PAS administrator is able to specify attribute types and values of his own choice on a per user or per role basis. These will be inserted in any PACs produced for the user, depending on what the user asks for. In a PAC request a user may specify:

- **Defaults**: In this case he gets back all the privileges attached to his name and all the privileges attached to his default role.

- **A role name**: In this case he only gets back the privileges attached to that role name.

- **Specific attribute types**: In this case he only gets back the specific privileges corresponding to these types.

**Miscellaneous (Audit Identity)**

The PAC owner’s audit identity appears as a miscellaneous PAC field. No other miscellaneous fields are supported in the current release. If a specific Audit Identity for a user has been specified in the PAS database, then this is used. Otherwise the value defaults to the Authenticated Identity supplied in the PAC request. The value is placed in audit records by the Audit Facility.

**Time Periods**

Identifies one or more time periods outside which the PAC is invalid. In SESAME a fixed set of time periods can be specified for all PACs, but different times cannot be specified for different users. This check supplements the PAC expiry check discussed previously.

### 5.1.3 Signature

See Table 3.

### 5.2 Selecting and Extracting Privileges and Controls

At the initiator end, privilege attributes and the controls over the PAC’s use that appear in the default PAC are chosen on the basis of the users authenticated identity and the role name. This may result in any valid combination of the attributes identified above, along with a number of other controls. In addition, both for pre-defined attributes and for administrator defined ones, the application may request specific PACs that only contain a subset of the permitted attributes (principle of the least privilege). As mentioned above, the target application server uses the Access Identity and Role attributes in its access control logic to grant or deny access to the application. Target applications can add their own more sophisticated controls based on the incoming attribute values.

### 6. Comparison to X.509 Certificates

Although this work started in the late 80’s and the certificates, defined in [11], did not allow at that time for an implementation of an authorization model, they do now. The extensions introduced in version 3 render it possible to define extra attributes upon which access control decisions can be based. However, in our view this approach suffers from some major disadvantages:

- **Trust**: In our scenario, we can distinguish between several security domains. Each domain is responsible for handing out its PACs and thus controlling the access to its applications. In our view, in the case of X.509 based certificates, we see a need for a centrally trusted CA that decides what rolenames to place inside the certificate. If this case is not workable (many companies may be reluctant to do this), then it is necessary to use many different certificates, thus introducing complex certificate management problems.
Table 3. Contents of the PAC’s Signature

<table>
<thead>
<tr>
<th>Value (of Signature)</th>
<th>The PAC is digitally signed using a public key algorithm (RSA with a key-length of 512 bits), by applying a message hashing algorithm (MD5) to the PAC body and encrypting the resulting value using the private key of the PAS. It could be argued that in view of current developments ([16, 2]), this should be replaced by RSA with a key-length of 1024 bits and RIPEMD-160 [3] or SHA-1 [8] as the hashing function.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm Identifier</td>
<td>This field identifies the algorithms used above, but it is not used in SESAME (instead the same value as in the PAC body is used).</td>
</tr>
<tr>
<td>Certificate Information</td>
<td>This field may be provided to help target application servers receiving the PAC to validate the PAC signature. In the SESAME implementation, its presence is configurable by the PAS administrator. If present then it contains the Directory Certificate for the PAS’s public key, signed by a Certification Authority (CA).</td>
</tr>
</tbody>
</table>

- Revocation: If a subject’s privileges change (e.g. due to a change of job), his certificate will need to be revoked, despite the fact that his key-pair has probably not been tampered with. X.509 based certificates have become very popular, especially in WWW security, but the problem of revocation has not been successfully addressed. PACs have a limited lifetime, and thus they do not suffer from the same problem.

- Expensive: Whether you buy the software to generate X.509 compliant certificates and pay a per certificate license, or whether you buy your certificates directly from a well known Certification Authority (e.g. Verisign), the average cost will be around $100 per certificate.

- Delegation: The solution involving PACs allows for a nice delegation feature, where the privileges can be passed on temporarily. This is impossible in the case of X.509 certificates (it is unthinkable to pass on the associated private key).

SESAME does use X.509 certificates, but only for identification (or authentication) purposes. We think that the mix of X.509 certificates and attribute certificates offers a lot of benefits.

7. Secure Intranet

In collaboration with the Queensland University of Technology (QUT), we have been implementing a secure Intranet based on the SESAME technology. Our situation was the following:

- We are an academic institution, and willing to share our computing resources.
- We have a lot of PCs with Linux, where the local user usually has the root password.
- Most users have an excellent knowledge of Unix.
- There was a clear need for using NFS to export file systems. As an example, our mail is read through NFS exported directories from the central mailserver.

It is clear that this situation was in urgent need of some security. Due to the many Unix bugs and features, it was fairly easy to abuse the local root privileges on other machines. A user could read our Dean’s electronic mail, try to get access to a professor’s account to search for examination questions or (even better) results.

To use SESAME, it was necessary to rewrite the popular Unix programs (telnet, rtools, ftp, NFS). The current situation is that all but one of these programs (ftp is the exception) has been adapted and tested in the Linux environment. With these tools, it is possible to avoid a local root
user from abusing his privileges as the privilege attribute server will decide who gets the global root privileges in our domain. Also, because of the extensive logging possibilities in SESAME, it is possible to trace any possible intruder and to act more rapidly upon an intrusion attempt. An extra benefit now is that all communications on our Intranet are also encrypted so that we are also protected against someone trying to sniff our local Ethernet.

8. Conclusion

We wanted to share our experience in writing an implementation of RBAC. Our work has shown that such a scheme can be realistically enforced. Much work has been done in all the standardization bodies (ISO, ANSI, ECMA, ...) and it is now up to the implementers to use this knowledge. Adapting the applications can be difficult, but it is usually straightforward when the GSS-API [14, 24, 17, 1] is adopted. All our applications are, or will be, available at [20].

For future work we are considering a solution where SSL (Secure Sockets Layer) [9] and SESAME would be used together to create Virtual Private Networks (VPNs) over the Internet. In this perspective, SESAME would add the possibility for non-repudiation and access control capabilities to the solution based on SSL.

9. Acknowledgments

The authors would like to thank all the people they have had the pleasure to work with on the SESAME project. In particular Michael Roe, Tom Parker and Denis Pinkas deserve a special mention for their design and analysis of the protocols.

References