

Providing X.509-based user access control to web servers*

A. Lioy, F. Maino

Politecnico di Torino - Dip. Automatica e Informatica

corso Duca degli Abruzzi 24 - 10129 Torino (Italy)

phone: +39-0115647021, fax: +39-0115647099

e-mail: lioy@polito.it, maino@polito.it

Abstract

This paper describes an access control model based on X.509v3 certificates for user authorization on HTTP servers secured by SSL.

The authorization model presented is based on the concept of *authentication roles*, that are the handlers that identify a single certificate (or a group of them) inside the access control list (ACL). The separation between authentication (role mapping) and authorization makes simple to write ACLs, and at the same time provides enough flexibility to filter authorized certificates.

The authorization model is presented, critically analyzed, and compared with the basic HTTP authentication scheme. Finally the implementation of this authorization model is given. It has been developed as a module for the *Apache-SSL* HTTP server, the SSL version of the most widely used WWW server on Unix platforms.

Keywords

Role based access control, public key certificates, network and web security

1 INTRODUCTION

The work presented on this paper has been developed within a project for the secure exchange of documents in the Italian public administration (Lioy *et al.* 1997). The need to share and distribute documents between the departments and the citizens, drove us to the development of an information system based on a standard WWW server, secured by SSL (Freier *et al.* 1996), that acts as a safe repository for signed and encrypted documents. The main problem here is to strongly authenticate users in order to authorize them to act as a document reader or as a document author.

While the user authentication can be provided at the protocol level through

*This project has been supported by the Italian government under contracts IPA-Demostene and MURST40%

the SSL option (*client authentication*) that compulsory requires a valid X.509v3 certificate to the client, the user authorization must be provided at the application level by the WWW server. Nevertheless, the most common HTTP servers still don't support methods for strong user authorization.

The only widely used standard authorization method is the *HTTP basic authentication scheme* that is based on username and password. This approach provides a very low security level also when used together with a channel security protocol such as SSL, because authentication relies on passwords that could be guessed by malicious third parties or improperly managed by users. Some SSL secured HTTP servers offer few configuration directives that allow filtering between valid certificates presented from the client, but what is needed is a flexible mechanism that allows a "per-directory" control on accessing documents.

Thawte Consulting proposed the *Strong Extranet*[®] system (Shuttleworth 1998), that embeds in each X.509v3 certificate a number of *SubjectAltName* private extensions. Each extension maps a relative identity (e.g. the student member number of a university, or the customer number of a bank) on the absolute identity of the certificate holder. The relative identity becomes the principal for authorization, and each institution can easily check the relative identity of its own members, regardless the other identities, absolute or relative, carried by the user certificate. The biggest drawback of this proposal is that each time a user needs a new relative identity he needs to get a new certificate from its CA. This can be a serious problem in first-click contacts between potential customers and Internet on-line service providers. Furthermore, a company that wants to start authorization check based on relative identity needs a special agreement with the CAs that distribute certificates to its users. Both this problems are, of course, less relevant in an Extranet environment, but make the Strong Extranet[®] system difficultly suitable for wide open community of users.

In order to provide access security to distributed computer applications, a big effort has been done in the research area known as RBAC (*Role-based Access Control*). RBAC models (Sandhu *et al.* 1994) (Jaeger *et al.* 1995), aim to define a framework in which users are mapped on roles that become the principals for authorization. Tari and Chan defined a very interesting and complete RBAC model (Tari *et al.* 1997) for intranet security that identifies two separated hierarchies of roles: the *local role* hierarchy describes permission of objects to use individual server's resources, while the *global role* hierarchy relates to resource accessibility across the whole intranet. The model addresses very well the problem of access control in an intranet environment, but in order to be implemented in the existing web servers requires in-depth changes to the architecture of the server.

In the definition of our access control model we followed a very pragmatic approach: we need a system for the secure management and distribution of documents within communities like the employee of a municipality or be-

tween the different departments of the public administration. The users of these communities are certified by separated, but interoperable, CAs within the ICE-TEL infrastructure (Chadwick *et al.* 1997), an European-wide public key certification infrastructure sponsored by the European Union (DG-XIII) within the *TELEMATICS for research initiatives*. This means that the principal for the authentication is the general purpose X.509v3 certificate held by each user of different communities, and that access authorization to a web server must rely on local roles, simply mapped on the general purpose certificate by the local webmaster. The mapping between roles and certificates, as well as the ACLs that state the access rights for each role, must use a syntax that is a natural extension of the configuration rules for the *Apache-SSL* HTTP server, the SSL version of the most widely used WWW server on Unix platforms. Nevertheless the model must be flexible enough to be extended and adapted to work within different architectures, such as the Netscape Enterprise web server or similar commercial products. It should be emphasized that the proposed authorization model, although based on the X.509 standard, doesn't require the support of the X.500 directory. We are planning to extend the model to use the secure LDAP (Lightweight Directory Access Protocol) protocol to retrieve the role definitions: this would allow to increase the scope of roles from a single server to the enterprise-wide domain.

The following sections describe the access control model and its implementation as an authorization module for the Apache-SSL server. After a short analysis of the basic HTTP authentication scheme, our authorization model is presented and critically analyzed. Finally a description of the main problems encountered on implementing the authorization model is given, along with directions for future work.

2 THE BASIC HTTP AUTHENTICATION SCHEME

The only authentication and authorization standard for the HTTP protocol is the basic HTTP authentication scheme. Despite of its name it is used not only for authentication, but also for authorization.

The principal for authentication is the pair *username/password* provided by the remote user that would like to access a protected document. The message exchange between client and server is as follows:

1. the client sends an ordinary HTTP request for a protected document
2. the server answers that access to that document is denied, sending an HTTP *unauthorized* message, with the name of the authentication realm to which the document belongs
3. the client asks the user for her/his authentication principal (username and password) for the specified realm
4. the client sends again the HTTP request for the protected document, together with the username and the password base-64 encoded

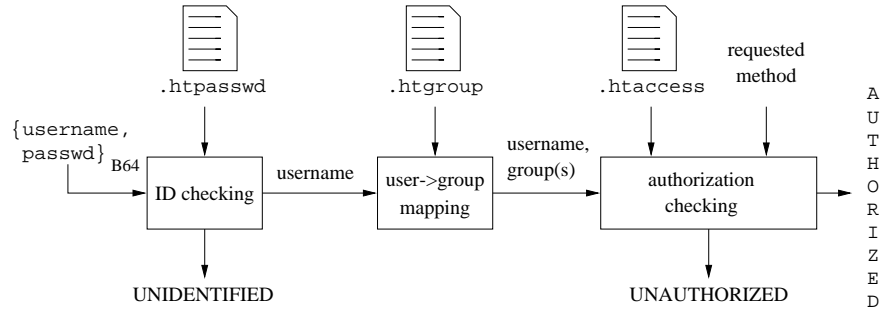


Figure 1 The basic HTTP authentication (and authorization) scheme

It should be noted that the base-64 encoding of username and password doesn't provide any further level of security to the password transmission. Only the SSL channel encryption grants secrecy and integrity when a password is transmitted over the network.

The received authentication principal must be checked by the server in order to identify the user and authorize the requested operation. This is done in three separate steps (Figure 1):

authentication: the username and the password are checked against the file `.htpasswd`, that contains pairs of valid username and password. If this step fails, the access to the protected document is denied because the user is unidentified

group mapping: the valid username is mapped on one or more groups described in the file `.htgroup`. This step would be absent if group authorization is not required

authorization: the username and, if present, the group identities are checked against the ACL `.htaccess`. The requested operation is performed by the WWW server only if the user is authorized to access the protected document with the requested HTTP method (GET, POST, PUT, DELETE or HEAD). If the user is not present into the ACL, an unauthorized message is sent back to the client and the access to the protected document is denied

3 THE PROPOSED AUTHORIZATION MODEL

The efforts in defining new proposals for public-key based infrastructure, such as the MIT SDSI (Simple Distributed Security Infrastructure) (Rivest *et al.* 1996) or the IETF SPKI (Simple Public Key Infrastructure) (Ellison *et al.* 1997), and works like the one of Blaze, Feigenbaum, and Lacy (Blaze *et al.* 1996) are based on the assertion that schemes like X.509v3, that require global certificate hierarchies, are both excessively complex and incomplete. The use

of an infrastructure that depends on global name spaces effectively requires formalization of many aspects, but it is possible to design real applications that rely on such infrastructure for the access control.

In our authorization model the principal for authentication and authorization is the X.509v3 certificate presented from the client when the protected document is requested. Table 1 shows the main elements of our authentication scheme in terms of access matrix model (Ford 1994). The user identity is related to the subject field of the client certificate and can be mapped over one or more *authentication roles*, that are the handlers that identify a single certificate (or a group of them) inside the ACL. The protected resources are the HTTP methods of access to a document: GET, POST, PUT, DELETE and HEAD. In order to better identify the remote user, some restrictions on other fields of the client certificate could be necessary. As an example, we could trust the identity of an employee only if the presented certificate has been released from his company CA.

User identity	Role	Type	Methods	Restrictions
/C=IT /O=Torino Municipality /OU=Town Council /CN=Valentino Castellani /Email=vale@torino.it	mayor	individual	GET, POST, PUT, DEL, HEAD	issuer=... size>=... valst>... valend<...
/C=IT /O=Torino Municipality /OU=Town Council /CN=([/]*) /Email=([/]*)@torino.it	counselor	group	GET, POST, PUT, DEL, HEAD	issuer=... size>=... valst>... valend<...

Table 1 Access matrix of the authorization model

This model maintains the ACL structure simple and, together with the use of regular expressions (REs) for the role-mapping of group of users, allows flexibility and efficiency in defining roles.

The proposed authorization scheme, as shown in Figure 2, has three steps (signature check, role mapping, and authorization check) that are explained in the following sections.

3.1 Signature check

The first step of the authorization scheme is performed from the WWW server at the begin of each SSL connection: the X.509v3 certificate presented from the client is accepted only if it has been signed from a CA that belongs to a trusted hierarchy. The list of trusted hierarchies is under the control of the webmaster,

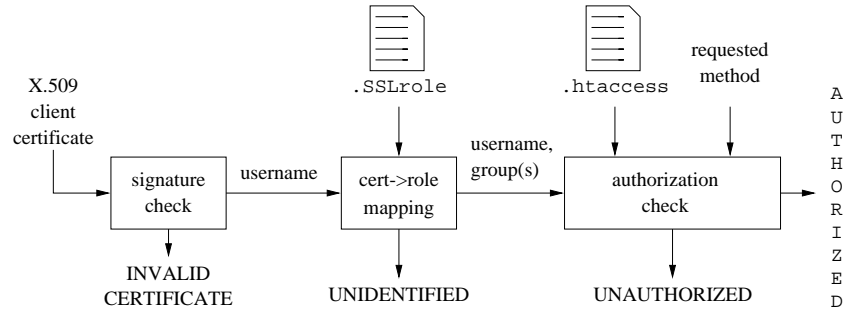


Figure 2 The proposed authorization scheme

and the user that publishes a protected document can only restrict the domain of trusted CAs. If the presented certificate is valid, the SSL connection goes on and the certificate is used as the principal for authorization.

3.2 Role mapping

In the second step of the authorization scheme the presented certificate is mapped over one or more authentication roles, in according to the roles definition file (`.SSLrole`). A different version of the file can be specified for each protected sub-tree of the filesystem and contains a description of the authentication roles in terms of fields of the X.509v3 certificates. A simplified version of the proposed syntax for role definition is given in Table 2.

```

role ::= rolename ":" role_def { operator role_def }
operator ::= "and" | "or" | "not"
role_def ::= cert_spec | certchain_spec | role_spec
cert_spec ::= "cert" [ not ] field_spec
certchain_spec ::= "certchain" [ not ] field_spec
role_spec ::= "role" [ not ] rolename
field_spec ::= field_name "=" | "!=" | ">" | "<" field_value
field_name ::= "subject" | "issuer" | "valst" | "valend" |
              "size" | "alg" | "exp"
field_value ::= valid value for an X.509v3 field, with REs
rolename ::= string
  
```

Table 2 Syntax of the proposed role definition

The role mapping can be done in terms of three directives:

- the `cert` directive acts on the standard fields of the X.509v3 certificate presented from the client

- the `certchain` directive acts as the `cert` one, but it applies on one of the certificates of the certificate-chain presented from the client. This allows a specific control on the CAs of the trusted hierarchy: for example, we can map a certificate on a role only if it has been released from a CA that is within the hierarchy of another trusted CA.
- the `role` directive allows the definition of an authentication role in terms of previously defined roles.

The directives `cert` and `certchain` act on the standard attributes of the X.509v3 certificates such as: the subject, the issuer, the start validity date, the end validity date, the size of the public key, the algorithm used to sign the certificate, and the exponent of the public key.

An example of authentication roles definition is given in Table 3. It should be noted that the use of REs allows the map of a group of certificates on a single role maintaining the file `.SSLrole` compact and easy to understand.

```

mayor: cert \
    "subject=/C=IT/O=Torino Municipality/OU=Town Council\
    /CN=Valentino Castellani/Email=vale@torino.it"\
    and cert "issuer=/C=IT/O=Torino Municipality\
    /CN=Torino Municipality CA"
counselor: cert "subject=/C=IT/O=Torino Municipality\
    /OU=Town Council/(.*)" \
    and cert "issuer=/C=IT/O=Torino Municipality/CN=CA"
mayor-assistant: ( cert \
    "subject=/C=IT/O=Torino Municipality/OU=Mayor Staff\
    /CN=Marco Rossi/Email=rossi@torino.it" \
    or cert \
    "subject=/C=IT/O=Torino Municipality/OU=Mayor Staff\
    /CN=Ugo Bianchi/Email=bianchi@torino.it" ) \
    and cert "issuer=/C=IT/O=Torino Municipality/CN=CA"
anybody: cert "subject=(.*)"
```

Table 3 An example for the file `.SSLrole`

3.3 Authorization check

In the last step of the authorization scheme (see Figure 2) the authentication roles of the presented certificate are checked against the ACL stored in the `.htaccess` file. The requested operation is performed if, for the requested method, one of the authentication roles matched from the client certificate is present in the ACL.

The proposed syntax for the entries of the ACL is shown in Table 4. The

`valid-cert` directive allows the access to the protected HTTP method to any of the roles mapped in the `.SSLrole` file. A further refinement between valid roles can be done using the `cert` directive followed by valid role names for the protected method. This directive is especially useful in order to share a single role-definition file between many protected resources with different access domains.

```
acl_entry ::= "require" acl_entry_type
acl_entry_type ::= "valid-cert" | role_directive | cert_directive
role_directive ::= "role" rolename {rolename}
cert ::= "cert" certname {certname}
certname ::= string
```

Table 4 The proposed syntax for the ACL entries

The proposed authorization framework acts as a placeholder for the basic HTTP authentication scheme. This is the reason for the first two lines of the sample `.htaccess` file, shown in Table 5, in which the type and the name of the authentication method are specified. The `AuthSSLRoleFile` directive specifies the name of the file used for the role mapping. The `require` directive, that identify valid roles, is placed inside a `Limit` directive that specify which are the protected methods. The root directory of the protected filesystem subtree is identified from the location of the file `.htaccess`.

```
AuthType Basic
AuthName SSL Client Authorization
AuthSSLRoleFile /services/httpsd/support/.SSLrole
<Limit PUT DELETE>
require role mayor-assistant mayor
</Limit>
<Limit GET>
require role counselor mayor-assistant
</Limit>
```

Table 5 An example for the file `.htaccess`

4 IMPLEMENTATION

An implementation of the proposed authorization model has been written as a separate module for the Apache-SSL HTTP server. We chose the Apache server not only because it is the most widely used WWW server for Unix platforms, but also for the complete modularity of the code. The server is

written as a simple core program in C that performs the basic HTTP tasks; all other functionalities are provided from separated modules invoked from the core.

We developed a module called `mod_auth_ssl` that is invoked just before the standard module that performs the basic HTTP authentication scheme. If the module finds the directive `AuthSSLRoleFile` for the requested document, the X.509v3 certificate presented from the client is used as the principal for authorization.

In order to maintain full compatibility with the basic HTTP authentication scheme, we use an Apache-SSL directive (`SSLFakeBasicAuth`) that forces the core module to fill the username and password fields of the received HTTP request with the certificate subject. In this way the core program doesn't follow the standard basic HTTP authentication scheme asking the remote user for username and password, but directly call the authorization module to perform X.509-based authorization.

Normally if the authorization based on the client certificate fails, the HTTP connection is refused. Nevertheless we designed a directive for the `.htaccess` file (`AuthSSLAuthoritative off`) that, if the X.509v3 role mapping fails, let the basic HTTP authentication scheme going on. In this case if the user provides correct username and password, he can get the document also if he doesn't hold an authorized certificate. This feature can be used, for example, to provide temporary access (with weak authentication) to users that still don't hold an authorized X.509v3 certificate.

5 OPEN ISSUES AND FUTURE WORK

The authorization model presented, that is based on the concept of authentication role, can be a valid support for access control on HTTP servers secured by SSL. The separation between authentication (role mapping) and authorization makes simple to write ACLs, and at the same time provides enough flexibility to filter authorized certificates. If the group of users that must be authorized is certified from a special purpose CA, is very simple to map the group on a single role and write an ACL that doesn't need any further modification in order to authorize new group members.

The implementation of the proposed authorization model still needs some test, especially in the interaction with the other authentication methods in which the authorization process can fall back if the X.509v3 authentication fails.

Future work is oriented to provide a scheme in which the definition of roles can be abstracted from the server level, and risen up to the intranet domain level. This will allow the control of user access to replicated services without the need of replicating the role definition. The integration of the proposed authorization model with secure LDAP (Lightweight Directory Access Protocol) servers, for example, could lead to an enterprise-wide definition of roles. More-

over, the introduction of the concept of protection domain derivation (Jaeger *et al.* 1997) could allow the interoperability between roles defined in different domains, allowing the use of roles defined inside other trusted domains.

The use of the described authorization model in a complex information system, such the ones of a public administration, will certainly provide a valid testbed for the whole system. Moreover, the valuable feedback from the user community will help us on improving the characteristics of the final implementation.

6 REFERENCES

- Blaze, M., Feigenbaum, J. and J. Lacy (1996) Decentralized trust management. *Proc. of the 1996 IEEE Symp. on Security and Privacy*, 164-173.
- Chadwick, D. W., Young, A. J. and Kapidzic Cicovic, N. (1997) Merging and Extending the PGP and PEM trust Models - The ICE-TEL Trust Model. *The Internet Society Symp. on Network and Distributed System Security*, San Diego, California.
- Ellison, C. M., Frantz, B., Lampson, B., Rivest, R. *et al.* (1997) Simple Public Key Certificate. *IETF Internet draft*.
- Ford, W. (1994) *Computer Communications Security, Principles, Standard Protocol and Techniques*. Prentice Hall.
- Freier, A.O., Karlton, P.L. and Kocher, P.C. (1996) The SSL Protocol (version 3.0). *IETF Internet draft*.
- Jaeger, T. and Prakash, A. (1995) Requirements of Role-based Access Control for Collaborative Systems. *Proc. of the 1st ACM Workshop on Role-based Access Control*, Gaithersburg, MD.
- Jaeger, T., Giraud, F. and Islam, N. (1997), A Role-based Access Control model for protection domain derivation and management. *2th ACM workshop on Role-based Access Control*, Fairfax, Virginia.
- Lioy, A., Maino, F. and Mezzalama M. (1997) Secure document management and distribution in an open network environment. *ICICS'97 - International Conference on Information and Communications Security*, Beijing, P. R. China, 109-117.
- Rivest, R. L. and Lampson, B. (1996) SDSI - A Simple Distributed Security Infrastructure. *SDSI - working document*
- Sandhu, R.S, Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1994) Role-based Access Control: a multidimensional view. *Proc. of the 10th Computer Security Application Conference*, 54-62.
- Shuttleworth, M. (1998) The Strong Extranet[®]: real-World Personal certification. *Thawte Consulting South Africa - whitepaper*.
- Tari, Z. and Chan, S.W. (1997) A Role-based Access Control for intranet security. *IEEE Internet Computing*, **15**, 24-34.

7 BIOGRAPHY

Antonio Lioy is an associate professor of computer engineering at the Politecnico di Torino. Professor Lioy holds a *laurea* (aka master) degree in Electronic Engineering *summa cum laude* and a Ph.D. in Computer Engineering from Politecnico di Torino. He is a registered professional engineer and a member of the IEEE and the IEEE Computer Society. His research interests are in the fields of computer and network security, and CAD of digital systems.

Fabio Maino is a Ph.D. student in computer engineering at the Politecnico di Torino. He holds a *laurea* (aka master) degree in Electronic Engineering and he is a registered professional engineer. His research interests are in the fields of computer and network security, with special focus on digital certificates and public key infrastructures.