Privacy matters in directories

30th April 2007

Jose A. Accino, University of Malaga, Central computing facility, 29071 Malaga
E-mail: accino@uma.es

Victoriano Giralt, University of Malaga, Central computing facility, 29071 Malaga
E-mail: victoriano@uma.es

Javier Masa, RedIRIS, Edificio CICA, 41071 Sevilla
E-mail: javier.masa@rediris.es

1 Introduction

Any organisation needs means for information dissemination about itself and its members, to provide access to services to its users, as well as collaborating with other organisations. On the other had, persons that belong to the organisation want to be found, and to find and contact others working in similar fields. Modern information technology facilitates this tasks, but, at the same time, the technological explosion has lowered the barrier to get information and invade individual privacy, thus producing an increase of privacy awareness both at personal and institutional levels.

This privacy concerns have produced restrictive laws concerning information dissemination in many countries[1]. The EU has produced a series of directives on personal data exchange and protection (e.g.: October 24th 1995 95/46/EC European Parliament and the Council of Europe Directive on the protection of individuals personal data and the free transfer of such data). Said directives have generated laws like the Spanish LOPD (December 13th, 15/1999) that mandates for consent from the individuals for the publishing of their personal data on its 6th and 11th articles.

Corporate directory services have been applied to this task for some time now. They offer a central repository of organisational data with standard interfaces. Institutions are at a crossroads created out of their need for information dissemination and people's rights to hide their personal data.

The easiest solution to this problem is total elimination of directory services, which is clearly not the best available, though some institutions have decided to go this route. We consider this not to be a solution but a bigger problem.

Another approach to the problem consists in totally blocking access to the directory from outside the institution. This is neither the solution, because it renders those persons willing to be found fully unlocatable, and it does not protect the right to privacy. Any internally developed application could publish personal data without individual consent.

Our approach to solve the problem gives individuals full control over the publishing of their personal data. Although, such control is always under institutional policy, which, in turn, should be defined according to the law.

Available information is not affected for normal institutional use, if access is granted to internally developed applications or properly identified persons, whom have the proper access rights according to institutional policy.

The directory manager defines the data sets that can be presented in the results of anonymous searches, according to institutional policy. This can be done, for example, using OpenLDAP Access Control Lists.

Users can then control access to their data, once the policy has been defined. Access control at the application level should be discarded, because it puts policy compliance in the hands of application programmers, making it very difficult to verify in organisations with multiple development teams.

The differential characteristic in our approach is the definition of data access policies at directory server level, which applies policy regardless of the access mechanisms be them direct server connection or application mediated.

In this paper, we will discuss the most relevant aspects of corporate directories, we will present and analyse the diverse security and privacy management policies, we will describe our solution and we will end with an use case of the solution.

2 Overview of corporate directories

The term directory is an ambiguous one as it can be used both for the information, the software/hardware system that processes said information or the client/server applications that use it. This leads to the conclusion that a Directory Service is a complex set of parts that cooperate to provide

a service.

2.1 Directory services

Classical paper directories are static, as they are printed at fixed intervals. On the other hand, electronic directories can be updated in real time, increasing their reliability. Also, classical directories are designed for searching on a fixed term, this is not the case with their electronic counterparts, were any stored information can be searched upon.

Access to classical directories cannot be controlled, anyone with physical access to the book get all published data. Thus, an individual's data is totally available or totally hidden. Access to data in electronic directories can be controlled according to different criteria. Although this is not a complete solution as anyone with appropriate privileges can produce a printed copy an distribute it, but anyway it means a higher level of security than in classical directories.

Electronic directories can also present the data according to the privilege levels of the person making the query, for example, showing personal exam results only the student that took the exam, or all results to the teacher that passed the test to a class.

Directories can be thought of as specialised databases adapted to their usage patterns and application areas[2]. The information stored in directories has a very low rate of changes which allows for optimisation for reads overlooking writes. The directory schema can be changed in response to changing needs in the organisation.

Electronic directories can be accessed by many differing applications, thus requiring a high level of standards compliance. This, in turn, confers a higher level of freedom on product selection to the administrators: The directory system can be changed without affecting client applications.

2.2 Standards. LDAP

The CCITT defined the X.500 standard in 1988, later adopted by ISO as ISO 9594 (Data Communications Network Directory, Recommendations X.500-X.521)[3].

The X.500 standard organises entries in a hierarchical way and is designed to allow for highly searchable and scalable big data volumes. The standard initially defined an OSI protocol for accessing the data (DAP, Directory Access Protocol). This has been superseded by a TCP/IP based lighter and easier to implement version of the protocol: LDAP or Lightweight Directory Access Protocol[4].

Client/server interaction in LDAP occurs in accordance to the following sequence:

1. Client initiates a session with the server. This session may be anonymous or the client may send credentials to start an authenticated one.
2. The client performs actions on the data. This actions may be searches, reads or updates.
3. The client closes the session ones it finishes operations.

LDAP directories follow the X.500 data model, organising stored information into data structures known as entries. Each entry describes an object, be it a person, a network node, a printer or an Internet domain name. Each entry is identified by an unique name called Distinguished Name (DN). DNs consist of smaller parts called Relative Distinguished Name (RDN), that describe the path that leads to the entry in the Directory Information Tree (DIT).

Object classes are general descriptions of object types. Entries belong to one or more object classes. The directory schema describes allowed object classes, their attributes, if these are required or not, and their format.

The LDAP standard defines primitives for access to and modification of entries stored in the directory:

- User defined searches
- Entry add
- Entry remove
- Entry modify
- Entry rename, i.e. changing the DN.
- Entry compare

2.3 Directory service as infrastructure

A directory service serving a disparate array of applications become a vital part of a system, as it provides uniform access to persons, resources and other system objects, thus, the directory is seen as a whole instead of a composite of independent parts.

The use of a directory service in applications eases their development and extend their capabilities. It is possible to develop a point to point videoconferencing application with user location and system capabilities assessment.

Such application with require several complex modules and would use proprietary protocols to communicate, which in turn will hinder interoperability with other similar applications. It is also possible to store system configuration and user location data in a central directory service and use standard protocols to retrieve the information. An application using this approach could be easily extended to find out the closest phone extension number to use as backup.

Benefits of a central directory are not only these, as information used by an application that does not rely on a central directory can only be accessed by such application. In such an environment each application manages and stores relevant information, and several of them could store the same data, thus leading to inconsistencies and redundancies.

Applications can use a common directory service through standard APIs that can be implemented in diverse platforms. This reduces the requirements for the applications and increases reliability.

2.4 Security and privacy in directories

Security of information stored in a directory is one of the main aspects in such systems. Some directories must allow for public access, but not all users may do all operations. The security policy determines who has which access levels to what information.

The directory must provide the basic capabilities to implement the security policy. First of all, a method for authenticating the user is required. Once the client identity has been established it is possible to allow or reject the requested operation.

Authorisations are often based on Access Control Lists (ACLs). This lists can be applied to objects and/or attributes of the entries stored in the directory. Users with the same permissions are usually grouped into security groups to ease the administration tasks.

Many organisations use of a central LDAP directory to store users data conflicts with the users right to privacy, which is granted by law in many countries (e.g.: the Spanish LOPD already mentioned above): some users do not like to see their data exposed to the public.

The problem could be solved by either giving control on displayed data to internally developed applications or by totally blocking external access to the directory. None of these is acceptable for promoting the use of collaborative inter-institutional applications and services. Thus, it becomes necessary to device a mechanism that allows users to decide which part of their personal data may be publicly accessible from the start.

3 Privacy policies

Corporate directories, in their own nature as central information repositories with standard access mechanisms, pose serious risks to privacy. These risks belong to three main groups: data exposure to external applications, privilege elevation in internal applications and, with special relevance due to the infrastructure role of the centralised directory, great difficulty for user direct control of the way their data are accessed.

The rest of this part will discuss the different protection schemes applied to directory services, analysing their pros and cons, and it will end with the description of the protection architecture we propose.

3.1 Firewalls

The obvious solution for protecting any kind of data stored in an organisation internal server is the use of a firewall based architecture, along usual corporate guidelines for limiting access to internal services from external networks.

This technique allows for isolating the directory from external unwanted accesses, but it works following all or nothing principles, which means that all internal applications are considered trustworthy. From a realistic point of view, this is not the case of any organisation, whichever its size: most applications deployed are not reviewed for compliance to minimal security and privacy respect criteria. Even imposing such standards to internally developed applications often reveals itself as a daunting task.

Moreover, as the access control is completely within critical elements of the security infrastructure, users have no means of accessing, lest modifying, the rules that govern access to their personal data.

Internal applications unwanted behaviour can be curtailed implementing firewall rules at the directory server, limiting access to parts of it depending on internal source addresses or requiring application specific identification rules.

However, this solution scales poorly as the number of applications using the directory increases, which is a desired target for any corporate directory service. And, at the same time, users are denied any control and knowledge of the rules governing access to their personal data.

3.2 Attribute release policies

Attribute Release Policies (ARPs)[5] are the answer to allow the users a greater control on how applications access their personal data stored in the corporate directory. An ARP defines which data can be revealed to which

applications for a given user. ARPs can be implemented using directory access APIs or by mediation of access services, as is the case in SOA based architectures (CORBA, WS, ...).

AS ARPs are sets of rules applied only to directory access and singled on the accessed entry, it is possible to allow users to know then and even participate in their definition. There are systems that permit such access through a web browser.

This solution still has the handicap of ill behaved applications that scape adaptation controls. This can be, for example, due to development previous to the model deployment or to inability to integrate with the adaptation level (API, service,etc) that applies ARPs. The need for the adaptation level breaks the principle of standardised access inherent to the directory, inducing an additional dependency that the directory service is designed to eliminate.

3.3 Operational ARPs

We have defined, developed and deployed the concept of **operational ARP** for solving the problem described above and to provide a reliable and scalable user managed personal data access control schema that respects access standards to directory servers. ARPs that apply to a given directory entry are stored within the entry and the server has mechanisms to enforce[6] them to any access that is done through the server standard mechanisms. This ARPs are compatible with any of the measures described above, which can and should be used for coarse grained controls, and can be applied to most LDAP servers available at present by means of careful configuration of access control parameters.

It should be noted that the use of operational ARPs allows the server to be open to external application access as control is applied inside the server and using standard access mechanisms.

4 schacUserPrivateAttribute

schacUserPrivateAttribute is an implementation of operational ARP based privacy management mechanisms described in the previous section. This proposal derives from the original implementation by the University of Malaga and RedIRIS that was included in the schema definitions recommended by the Spanish NREN to affiliated institutions. Later, the proposal was included in the European Academic Schema Harmonisation, SCHAC, and thus the name change. We are working to submit the proposal to the IETF to be considered as an operational standard for LDAP servers.

`schacUserPrivateAttribute` is an attribute used to specify access policies to the other attributes in a directory entry, specifying those attributes in the entry that should be blocked from access to their values.

4.1 Work flow

- The institution defines a list of Optional Private Attributes (OPA) that the user can define as hidden.
- The institution defines the rules, in the LDAP server access control list, that keeps privacy of such data without hindering communication with other applications and services.
- The user decides which attributes to hide adding their names to the values of schacUserPrivateAttribute.

4.2 Optional private attributes (OPA)

Each institution may define one or more lists of permited values for the set of optional private attributes. One of such lists might include:

- `telephoneNumber`
- `facsimileTelephoneNumber`
- `mobile`
- `postalAddress`
- `postalCode`
- `homePhone`
- `homePostalAddress`
- `mail`
- `labeledURI`
- `title`
- `description`
- `jpegPhoto`

There are two values with special meanings in addition to those defined by the institutions:

`all` denies access to all attributes in the APO list, allowing the user to block access to the whole set without the need to enter all of them one by one.

`entry` denies access to the whole entry.

4.3 Formal definition of schacUserPrivateAttribute

The attribute definition in an LDAP server scheme is as follows:

```
attributetype ( 1.3.6.1.4.1.25178.1.2.18
      NAME 'schacUserPrivateAttribute'
      DESC 'Set of denied access attributes'
      EQUALITY caseIgnoreIA5Match
      SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

The OID hierarchy is 1.3.6.1.4.1.25178 for TERENA, .1 for SCHAC and .2 for attributes.

The OID for irisUserPrivateAttribute, included in irisPerson object class is: 1.3.6.1.4.1.7547.4.3.2.11

4.4 Usage examples

The server will not return attributes `mail` and `telephoneNumber` by just including their names as values of `schacUserPrivateAttribute`

```
schacUserPrivateAttribute:  mail
schacUserPrivateAttribute:  telephoneNumber
```

The server will not return any value included in any of the attributes included in the APO list:

```
schacUserPrivateAttribute:  all
```

The server will not return the entry:

```
schacUserPrivateAttribute:  entry
```

5 Use case

The University of Malaga corporate directory implements the privacy control mechanisms described above, using an OpenLDAP server. This does not preclude the use of other servers. Preliminary research to use this principles to Red Hat's Fedora Directory Server is promising. Providers of other directory servers have confirmed that the approach can also be applied to their products.

5.1 Directory server configuration

Access to data is controlled by means of Access Control Lists (ACLs)[7], which allows both for type of access (identified, anonymous) and level of access (search, read, write, none) differentiation. `schacUserPrivateAttribute` together with corresponding ACLs allows for privacy control of personal data at the directory server without degrading communications with other services.

`schacUserPrivateAttribute` is multivalued and holds the names of those attributes the user decides to hide to anonymous searches. These attributes are a subset of the OPA list defined by the directory administrators.

The directory administrator defines the user controllable attribute list and the access levels using ACLs, that determine if the values a given attribute are returned in search results depending on the attribute name being present in `schacUserPrivateAttribute` values, the connection type (anonymous or identified) and the user performing the search in identified ones. The ACLs shall be defined with ascending granularity, i.e. finer grains should be first.

It is also possible to apply different policies to entries of persons with different institutional affiliations.

The following example show total access blocking of an entry:

```
[1] access to *
     filter="(schacUserPrivateAttribute=entry)"
     by * none
```

The following example show the policy that applies to persons with only student affiliation to the University:

```
[1] access to *
     filter="(&(eduPersonPrimaryAffiliation=student)(schaUserPrivateAttribute=all))"

     attrs=entry
     by * none

[2] access to *
     filter="(eduPersonPrimaryAffiliation=student)"
     attrs=entry,displayName,mail,telephoneNumber
     by * read

[3] access to *
     filter="(eduPersonPrimaryAffiliation=student)"
     by * none
```

Rule 1 blocks access to the whole entry, if the value of the privacy attribute

is *all*, the default for students.

Rule 2 allows access to those attributes of students that the University has decided to allow, if the person decides to allow access clearing the privacy attribute.

The following example shows access blocking to `mail` and `mobile` attributes, as well as the whole OPA set and the entry, for those persons whose main affiliation to the University is not student.

```
[1] access to *
     filter="(irisUserPrivateAttribute=entry)"
     by * none

[2] access to *
     filter="(irisUserPrivateAttribute=mail)"
     attrs=mail
     by * none

[3] access to *
     filter="(irisUserPrivateAttribute=mobile)"
     attrs=mobile
     by * none

[4] access to *
     filter="(irisUserPrivateAttribute=all)"
     attrs=mail,mobile
     by * none

[5] access to *
     attrs=displayName,mail,mobile
     by * read

[6] access to *
     by * none
```

- Rule 1 controls access to the whole entry.
- Rules 2 and 3 control access to attributes that the user may decide to hide from searches.
- Rule 4 lists all the user controllable attributes, so they are hidden from searches when **irisUserPrivateAttribute** is set to *all*.
- Rule 5 grants access to all displayable attributes in the entry if none of the previous rules are matched.

- Finally, rule 6 blocks access to non displayable attributes.

5.2 User management of attribute access controls

A web application has been developed to allow users to manage their personal data stored in the corporate directory as well as access permissions to such data. This application requires a double identification, it is placed in a restricted area of the Central Computing Facility web server, with access allowed only to persons registered in the corporate directory in possession of appropriate credentials. Access to this area is controlled with normal browser mechanisms. Then access to the application that manages personal data requires the user to reidentify, to prevent access on unattended workstations.

This application allows users to set the values of the privacy attribute through a simple web from.

References

[1] F. Lopez-Carmona, International data transfer and regulations. In: Proceedings of I Advanced EuroCAMP. Available at http://www.terena.org/activities/eurocamp/october06/programme2.html

[2] C. Severance. Could LDAP be the next killer DAP? In: Computer, Volume 30, Issue 8, Aug. 1997, Page(s):88 - 89

[3] ITU-T. X.500: Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services

[4] M. Wahl, T. Howes, S. Kille. RFC 2251: Lightweight Directory Access Protocol (v3)

[5] W. Hoehn. Attribute Release Policies: The Technology of Privacy. In: Proceedings of CAMP 2004. Available at http://www.educause.edu/LibraryDetailPage/666?ID=EAF0448

[6] M. Wahl, A, Coulbeck, T. Howes, S. Kille. RFC 2252: Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

[7] V. Hassler. X.500 and LDAP security: a comparative overview. In: IEEE Network, Volume 13, Issue 6, Nov.-Dec. 1999 Page(s):54-64