Global Information Assurance Certification Paper

Overview & Security Aspects of the-

Lightweight Directory Access Protocol (LDAP)

Louis R. Brand
April 17,2001

SANS Security Essentials
GSEC Practical Assignment
Version 1.2b

Introduction

The need for directory standardization has been recognized for many years within the computer industry. In the past, directories were application specific, yet the growth of network based applications has fueled the demand for applications requiring the ability to easily access information that normally resides in such data-stores. One such driver of standardized directory data access is in support of business cooperation that occurs via the Internet. The development of the Lightweight Directory Access Protocol (LDAP) has enabled and greatly advanced such implementations. This paper will explore this technology and its benefits in addition to its IT security capabilities and challenges.

Overview

How often has the need arisen to publish static information that might be used by end-users and/or applications? The following are a few examples of how this capability would be useful

- Security Advantages:
  - Quickly add, modify and remove user privileges
  - Repositories for certificates & associated public key information
- Ability to reveal directory based information based upon user authentication levels
- Single System Sign-On to eliminate the need for managing multiple userid and passwords.
- Workflow Management
  - Department hierarchies centrally defined to facilitate transfer of work items

Though there have been platform specific methods to support user administration such as SUN's Network Information System (NIS) and MS Windows' concept of a domain controller, LDAP is the only vehicle for a set of vendor and platform independent directory services that can support a heterogeneous client (TCP/IP and in the future, UDP) computing infrastructure. The need for network enabled directory services is being driven by the ever-increasing diversification of this infrastructure from the operating system, database (relational and other), the application layer services such as ftp and http. In addition to the end-user applications such as SAP and Notes that use all of the facilities of the underlying computing infrastructure.

LDAP has its roots in the International Organization of Standardization's (ISO) efforts to create a standards (not a vendor de facto or Internet standard) based directory and a supporting protocol suite. Of course, like any standardization effort especially one

compounded by worldwide membership, this is typically a slow effort that more often than not results in a complex conceptual model. ISO standards cover a wide spectrum of disciplines. Many of us are aware of the ISO 9000 suite of quality standards and possibly the ISO 14000 standards involving environmental controls, but they encompass much more than that. Also, in December 2000, ISO finalized an IT security standard: ISO 17799. Internally, ISO's standards process is: recognized need for a standard->Working Draft-> Draft Proposal->Draft International Standard->International Standard. During this time, technology and vendor implementations march on! Immediate real-world requirements force implementers to create solutions that: i) cannot wait for the final design, ii) cannot support the complexity of the model. Also, vendor developed proprietary solutions can insure a captive customer base as opposed to an open standard. ISO's X.500 standard, which was jointly adopted by ITU (a standards organization specializing in communications), is such a directory standard released in 1988. For reasons beyond those mentioned herein (required using an OSI/ISO communications stack and other OSI protocols rather than TCP/IP) and a poorly accepted API), X.500 was not widely embraced.

As opposed to the development of ISO standards, the Internet Community's "standards" evolution is much less confining. A technical body, known as the Internet Activity Board (IAB), oversees the development of the Internet protocol suite. There are working groups that focus upon specific areas, but anyone can propose a new standard which is initially documented as a Request For Comments (RFCs). In addition, RFCs can also be of an experimental or informational nature.

This latter pragmatic approach was used to develop a published protocol that could access a directory using TCP/IP and without incurring the overhead of X.500's Directory User Agent but yet did not implement all the feature of X.500. This work was spearheaded by the University of Michigan's IS department. These personnel then went to work for Netscape which soon unveiled LDAP V1.

X.500 & LDAP Overview

Introduction

While it might be construed that LDAP has solved all the real world problems that ISO's X.500 standard and subsequent vendor implementations did not, this is not true. What has happened is that although LDAP has fueled the implementation of networked directory services, the growing complexity of the applications has caused the LDAP effort to strive to implement the functions first in X.500. Also, the LDAP standard requires LDAP to have the ability to interface to an X.500 based directory server. As you can see, LDAP

and X.500 are intertwined. Therefore, we'll cover some X.500 concepts to give the reader further insight.

Overall, LDAP is targeted towards the implementation of a single directory server rather than a distributed environment as described and supported by X.500. To get around this failing, LDAP supports the concept of referral; if a client's request cannot be satisfied (at all or completely) it will be given the URL of other servers that may be able to support the request. (The definition and application of LDAP URL's is defined in RFC 2255.)

Another limitation is obviously replication and also to a large degree overall administration. There is no method defined for updating or replicating information between LDAP servers. Though LDIF (RFC-2849, the LDAP Data Interchange Format) has been defined to allow the import and export of a directory's contents or changes in flat file format, it does not come near to what is required to facilitate a distributing directory changes. Also, within a distributed directory environment, the ability to "join" the records of a directory with another's based upon a common directory "key" is useful in generating additional data relationships that are not inherent to a single database (in this case, a directory). But, this too is something that is not currently possible with LDAP.

LDAP Schema

Accessing LDAP information requires an understanding of its schema. This schema is comprised of two parts: the *Directory Information Base* and the *Directory Information Tree*. In addition, management tools support modifying the directory's schema so long as items such as naming conventions and facets defined as required are maintained.

Directory Information Base (DIB)

The DIB defines objects, which are known as *entries*. Each entry is identified by a globally unique *Distinguished Name* (whose syntax is described in RFC#2253*)*. An entry is defined by *attributes* (usually more than one) and each attribute is comprised of a type and value (single or multiple). One mandatory attribute is an *Object Class*, which defines the types of attributes the entry must and/or may have. RFC 2256 defines both mandatory and optional Object Classes and attributes that an LDAP directory server needs to support. These attributes and object classes are themselves derived from various X.500 standards

Directory Information Tree (DIT)

The relationship of the entries (again, entries are defined by the DIB) is described by the hierarchical tree structure of the DIT. Beginning with the root node, the tree is descended where each level of the DIT is known as the *Relative Distinguished Name* (RDN).

(Again, the DN is the path traversed to get to a specific RDN.) An example of descending a DIT for an organization would be:

- 'o'- organization name
- 'ou'- organization unit name
- 'title'- title, such as vice president
- 'givenName'- first name (not the surname or middle name)

Which would result in the following Distinguished Name: {o=SANS, ou=Education, title=Vice President, givenName=JrHeadCheese ..

As mentioned, RFC 2256 includes the definition of object classes and attributes that the LDAP compatible directory *must* or *should* support. Though this defines a basic underlying schema of the directory server database, LDAP is an access protocol; it has no concern for how the directory's data is physically represented. This is a good reason why you won't find directory server management as part of the defined LDAP API (described below). LDAP servers can be implemented upon proprietary databases (Lotus Notes/Domino), relational databases or B-Tree formats, not simply object databases. LDAP schemas are being developed to support a myriad of diverse applications requiring information to be published and accessible, for example:

- White Pages (RFC 2218)- Internet email and telephone directories are an area where exchange (and therefore agreed to schemas) of data is required.
- CORBA (RFC 2714)- Definition of a directory structure specific to supporting the Open Group's platform independent *Common Object Request Broker Architecture*. CORBA's goal is to facilitate worldwide network based object communication.
- Directory Enabled Networks - Creation of a common directory for storage of network equipment and protocol configurations.
- Calendars (RFC 2739)- Definition of an object class and attributes to create a common user calendar infrastructure.
- PKI (RFC 2587)- Definition of a schema to allow exchange of inter-enterprise public key information.

LDAP Application Programming Interface (API)

The definition of a relatively simple programming interface, abstracted the interface from the underlying physical representation of the data. This is what has fueled the growth of LDAP. The LDAP API is supported in both the C (RFC 1823) and Java (JNDI- Java Naming and Directory Interface) languages in addition to Perl (script). It embodies the capabilities defined by RFC 2251, the Lightweight Directory Access Protocol (v3). The following is a summary of the operations supported by the API. In addition to these, there are others that support real world computing events such as an Unsolicited Notification which, for example, are used to inform the active LDAP client that the server is about to enter maintenance mode.

LDAP API Operation Summary:

- Open- List of hostnames or IP addresses of the target LDAP Server(s). Connection attempts are executed sequentially until one is successful.
- Bind (and Unbind)- Used to authenticate a client to the LDAP server. Three types of bind are supported, which are explained below in the *LDAP Security Topics* section.
- Search- includes a filter capability. Returns matching entries for each requested attribute. The support of wild cards allows one to simulate the ability to list the children of an entry.
- Modify- A set of operations that provide the capability to modify an existing LDAP entry.
- Add- the ability to add entries to the directory. If necessary, the add operation will create the attribute.
- Delete- the ability to delete entries from the directory.
- Modify DN – the ability to change Distinguished Names
- Abandon- the capability to discontinue an operation that is in progress.

LDAP Security Topics

In this section, we'll explore a diverse set of topics that fall under the umbrella of security.

1. The security capabilities of LDAP (V3).
2. The application of LDAP to support a security infrastructure.
3. A directory's inherent security exposures.

LDAP (V3) Security Capabilities

Similar to the evolvement of so many other computing technologies, the security aspects of the LDAP standard were late in getting their due. Currently, the basic security model is simple yet it allows the implementation of some powerful technology. During the LDAP client's establishment of a session with the server (a *bind* operation), authentication is negotiated of which there are three possible levels:

1. No Authentication- This mode of operation would be applicable to access to a public directory.
2. Simple Authentication- In this mode, the contents of the bind API's password parameter would be sent in clear text. The recommendation is that clear-text password not be sent without some form of encryption provided by a lower layer protocol.
3. SASL- The *Simple Authentication and Security Layer* mode allows the use of any method or *mechanism* defined by the SASL framework. Although SASL allows the selection from a half dozen security mechanisms, Secure Sockets Layer

(SSL) and its successor TLS (RFC 2246- Transport Layer Security) are the most widely accepted for use with LDAP V3.

Applying LDAP to Enable A Security Infrastructure

The ability to publish and more importantly control access to information appropriate to directory services (information with a high read to update ratio) is of critical importance. Our growing networked world requires easy and secure access to information that often is the crown jewels of the enterprise. The following are a few examples of where LDAP is applicable.

Single System Sign-On- The ability for the applications and operating systems of the typical heterogeneous computing environment to authenticate users against a single directory would support the implementation of single system sign-on. Applications such as SAP R/3 are moving towards this; an imminent release will provide a portal into all of the applications that comprise an enterprise's typical SAP environment: Human Resources, Finance, Time-Keeping, Warehousing, Production Planning, etc.. SAP will interface to an LDAP compliant directory to provide user authentication for these various SAP applications.

User Administration- A perplexing problem for the typical enterprise environment is how to add and remove user system privileges quickly. The desire to add users quickly is a matter of productivity; when someone moves into a department or new work role they should have the correct systems access that their job requires. But more importantly is the need to quickly remove user rights to avoid potential systems abotage from dismissed or disgruntled employees.

Public Key Infrastructure (PKI)- One of the basic requirements of a PKI infrastructure is the maintenance of User Certificates. User Certificates contain an individual's public key together with additional identifying data. This certificate is signed/encrypted by a Certificate Authority (CA) (such as VeriSign) who guarantees that the certificate is valid (it has not been revoked, neither public nor private key has been compromised and the user's identity is also valid). This provides two benefits, the CA guarantees: the public key is authentic and it can be associated with the specific user. CAs such as VeriSign support the delivery of certificates to LDAP based directory systems.

Security Exposures

Technological advances usually also enable detractors and LDAP is not immune to this. Publishing information in a directory certainly increases its availability to unauthorized

scrutiny. Prior to the establishment of a directory, this same information would have been scattered helter-skelter between application specific directories and/or embedded within the application where it would have had some inherent protection. Now when an enterprise becomes directory enabled, additional security exposures are created. Utilizing LDAP's available security capabilities together with good security planning will minimize these exposures. The following is a discussion but not an exhaustive list.

Denial of Service Attack against an enterprise's directory server can be quite catastrophic. The applications that rely upon this data might no longer function or would certainly operate at some level of reduced function.

Man in the Middle Attack- An LDAP client could receive data whose source is not the assumed LDAP directory server.

Confidentiality of Data- Directory based data can be the jewels of the enterprise. Insuring that this data remains confidential, its availability commensurate with the access authenticity and uncorrupted will take on strategic importance.

Conclusion

Popular web browsers have supported LDAP for a few years and there are a multitude of LDAP compliant directories available in the public space; many of these directories are X.500 "back-end" installations with LDAP interfaces. Web servers such as Apache, Microsoft's IIS, Netscape and others all support external user authentication via LDAP. Finally, there are directory server offerings from vendors such as: IBM/Lotus (Domino) and I-Planet. Overall, LDAP implementations are quite widespread in the computer industry.

Though there are technical and security issues that remain in the evolution of directories especially as LDAP's capabilities evolve to meet increasingly complex requirements, there is no doubt that the growing ubiquitous networked world will continue to push this envelope. Securing an ever-growing amount of business critical information within this environment is surely a major challenge security professionals will face.

References

David Goodman, Colin Robbins. "LDAP- Moving Forward Frequently Asked Questions." July 2000. URL: http://www.nexor.com/info/LDAP-FAQ-23.htm

David Goodman, Colin Robbins. "LDAP- Moving Forward RFCs & Internet Drafts." September 2000. URL: http://www.nexor.com/info/LDAP-RFCs.htm

David Goodman, Colin Robbins. "LDAP- Moving Forward LDAP Business Applications and Scenarios." September 2000.
URL: http://www.nexor.com/info/LDAP-Apps/LDAP-Apps.htm

Good, G.. "Request for Comments:2849." The LDAP Data Interchange Format (LDIF) – Technical Specification. June 2000. URL: ftp:// ftp.isi.edu/in-notes/rfc2849.txt

T. Howes, M. Smith. "Request for Comments: 1823." The LDAP Application Program Interface. August 1995. URL: :// http://andrew2.andrew.cmu.edu/rfc/rfc1823.html

T. Howes, M. Smith. "Request for Comments: 2255." The LDAP URL Format. December 1997. URL: :// ftp.isi.edu/in-notes/rfc2255.txt

Jones, Vincent. MAP/TOP Networking. New York City: McGraw-Hill Book Company, 1988. 60-64.

Rose, Marshall. The Open Book. Englewood Cliffs: Prentice Hall, 1990. 446-458.

Wahl, M. et al. "Request for Comments: 2829." Authentication Methods for LDAP. May 2000. URL: :// ftp.isi.edu/in-notes/rfc2829.txt

Wahl, M.. "Request for Comments: 2256." A Summary of the X.500(96) User Schema for use with LDAPv3. December 1997.
URL: :// www.cis.ohio-state.edu/cgi-bin/rfc/rfc2256.html

Wahl, M. et al. "Request for Comments: 2251." Lightweight Directory Access Protocol (v3). December 1997. URL: :// ftp.isi.edu/in-notes/rfc2251.txt

"Introduction to Public-Key Infrastructure"
URL: http://www.iplanet.com/developer/docs/articles/security/pki.html