A trusted Network Model using the Lightweight Directory Access Protocol

D. C. VASILIADIS[1,2], G. E. RIZOS[1,2]
[1]University of Peloponnese
Department of Computer Science
and Technology
GR-221 00 Tripolis
GREECE

E. STERGIOU[2], S. V. MARGARITI[2]
[2]A.T.E.I. of Epirus
Department of Teleinformatics
and Management
GR-471 00 Arta
GREECE

Abstract: - Lightweight Directory Access Protocol (LDAP) is an Internet protocol that email and other programs use to look up information from a server. LDAP has been recently identified as an efficient technology for accessing common directory information. Moreover, LDAP is not limited to contact information or information about people, but it is used to look up encryption certificates, pointers to printers and other services on a network providing a "single sign on", where one password for a user is shared between many services on an intranet, or even on the Internet. Thus, LDAP provides an extendable architecture for centralized storage and management of information that needs to be available for today's distributed systems and services. In our paper we analyze a network model in the context of providing trusted information between a central database and one or more servers in the network. In the proposed network schema a server manager is coupled between the central database and servers. The server manager provides trusted communication by transmitting configuration information between the central database and servers in single communication channels.

Key-Words: - Lightweight Directory Access Protocol, Client/Server, Domain Name Service, Dynamic Host Configuration Protocol

1 Introduction

Nowadays, new applications require distributed computing implementations, relying on networked computer systems. This kind of applications operates with many network objects like computers on the same local area network (LAN), within a corporate intranet, or even on the Internet, where much of this information can be shared among many applications. Some of this information must also be protected to prevent unauthorized modification or the disclosure of private information.

Thus, LDAP is an appropriate method for any kind of directory-like information, where fast lookups and less-frequent updates are the norm. As a protocol, LDAP does not define how programs work on either the client or server side. It defines the "language" used for client programs to talk to servers or programs to talk among servers. On the client side, a client may be an email program, a printer browser, or an address book. The server may speak only LDAP, or have other methods of sending and receiving data, where LDAP may just be an add-on method.

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet these needs. It was designed at the University of Michigan to adapt a complex enterprise directory system (called X.500) to the modern Internet. X.500 is too complex to support on desktops and over the Internet. On the other hand, LDAP defines the de facto standard method for accessing and updating information in a directory, much the same as the Domain Name System (DNS) is used for IP address look-up on almost any system on the Internet. LDAP is already gaining wide acceptance as the most efficient directory access method of the Internet and is therefore also becoming an increasingly important driving accessing method within corporate intranets and on the Internet. It is being supported by a great number of software vendors and is being incorporated into a growing number of applications.

At first we present the Directory Client/Server system. Then, we explain a novel trusted network model. Finally we analyze the authentication method and provide the concluding remarks.

2 A Directory Client / Server System

Lightweight Directory Access Protocol (LDAP) is a set of open protocols used to access centrally stored information over a network. It is based on the X.500 standard for directory sharing, but is less complex and resource intensive.

Like X.500, LDAP organizes information in a hierarchal manner using directories. These directories can store a variety of information and can even be used in a manner similar to Network Information Service (NIS) [1], enabling anyone to access their account from any machine on the LDAP enabled network.

In many cases, however, LDAP is used simply as a virtual phone directory, allowing users to easily access contact information for other users. But LDAP is more flexible than a traditional phone directory, because it is capable of referring to other LDAP servers throughout the world, providing an ad-hoc global repository of information. Currently, LDAP is more commonly used within individual organizations, like universities, government departments, and private companies.

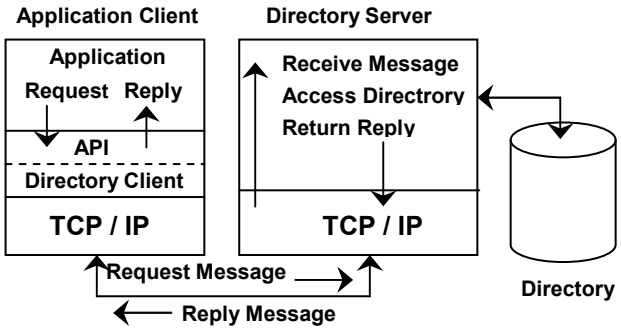

Fig.1. Directory Client / Server System

LDAP is a client-server system. Directories are usually accessed using the client / server model of communication. An application that wants to read or writes information in a directory does not access the directory directly. Instead, it calls a function or application programming interface (API) that causes a message to be sent to another process. This second process accesses the information in the directory on behalf of the requesting application. The results of the read or write are then returned to the requesting application (Fig. 1).

The server can use a variety of databases to store a directory, each optimized for quick and copious read operations. When an LDAP client application connects to an LDAP server, it can either query a directory or attempt to modify it. In the event of a query, the server either answers the query or, if it can not answer locally, it can refer to an LDAP server which does have the answer. If the client application is attempting to modify information an LDAP directory, the server verifies that the user has permission to make the change and then adds or updates the information. Configurations of OpenLDAP 2.0, an open source implementation of the LDAPv2 and LDAPv3 protocols are used (Fig. 2).

A solid foundation for a directory service infrastructure for the Internet was built by LDAPv3 [2, 5, 6, and 7]. Most of vendor implementations are based on this version or have most features of version 3 incorporated. But there is still room for enhancements, for example in areas of API support for other program languages, like Java. To define these standards, members of the Internet Engineering Task Force (IETF) work on and submit draft proposals that eventually might become Request for Comments (RFCs).

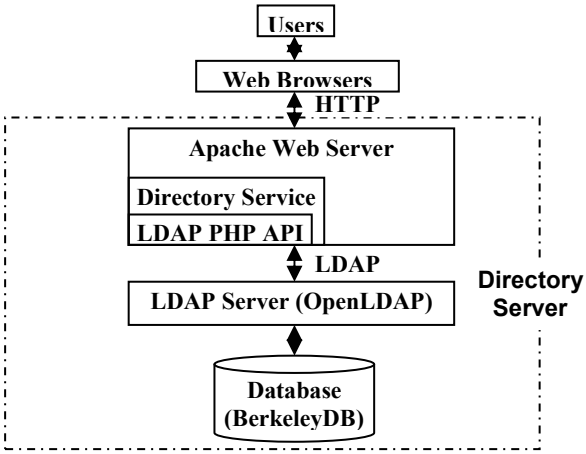

Fig.2. Directory System Architecture

A directory service is only one type of service that might be available in a client/server environment. Other common examples of services are file services, mail services, print services, Web page services, and so on. The client and server processes might or might not be on the same machine. A server is capable of serving many clients. Some servers can process client requests in parallel. Other servers queue incoming client requests for serial processing if they are currently busy processing another client's request. An API defines the programming interface a particular programming language uses to access a service. The format and contents of the messages exchanged between client and server must adhere to an agreed upon protocol. LDAP defines a message protocol used by directory clients and directory servers. There is also an associated LDAP API for the C language and ways to access LDAP from writing a Java application. The client is not dependent upon a particular implementation of the server, and the server can implement the directory however it chooses.

3 A Novel Trusted Network Model

In the proposed network model, each server directly communicates with the central database in order to obtain configuration information. The Fig. 3 illustrates an overall diagram of a conventional Transmission Control Protocol (TCP) / Internet Protocol (IP) network including a Domain Name Service (DNS) server, a Dynamic Host Configuration Protocol (DHCP) server, a Remote Authentication Dial In Service RADIUS Server, a MAIL Server, a World Wide Web WWW Server and a Central database. The DNS, the DHCP, and the RADIUS servers transmit requests for configuration information and send configuration updates to the central database on the network. On the other hand, the central database either transmits the requested configuration information back to each server or it stores the configuration updates received from each server. The proposed method provides information between a central database and other servers. Moreover, a server manager is coupled between the central database and all other servers. The server manager communicates configuration information between the central database and each one server in a single communication channel.

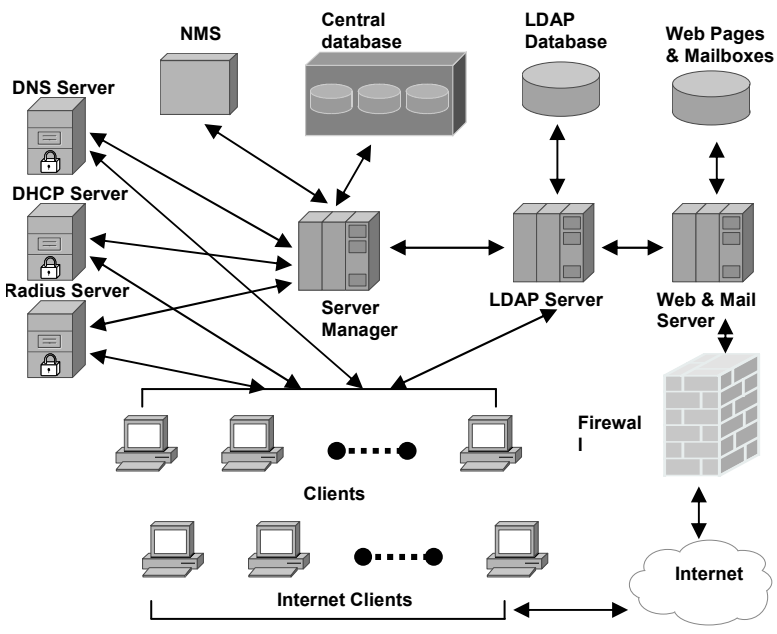

Fig.3. A network model

The use of a DHCP server simplifies the management and assignment of IP addresses to clients by eliminating the need for the network administrator to manually configure the network. When a client requests an IP address in order to communicate over the network, a DHCP server answers the request by providing network configuration information that was obtained from a database and dynamically assigning an IP address to the client. The DHCP server manages a range of client addresses and can pick any address from that range as long as it is not being used by a another client managed by that DHCP server. Since the address is dynamically assigned, the client can have a different IP address each time it logs on to the network. Along with the ability to dynamically assign IP addresses, a DHCP server also supports static IP address that have been assigned to one particular client on the network. Based on the configuration information received from the database, the DHCP server can automatically assign a specific IP address to a specific client. In the case of remote calls, clients are serviced initially by the Remote Authentication Dial In Service RADIUS Server.

A DNS server also simplifies the management of networks by translating domain names into IP addresses. Since the DNS server contains a list of domain names and their associated IP addresses, a host or client can be located through by its domain name rather than its IP address. Any given domain name could be associated with more than one IP address and any IP address could be associated with more than one domain name. A DNS server updates the domain name and IP address associations by periodically polling a central database containing configuration information for the network. When a new client is assigned an IP address by a DHCP server, the new configuration information is stored in the central database. Each DNS servers on the network poll the central database for configuration changes. If a new IP address was assigned to a client managed by a DNS server, the DNS server updates the domain name with the new IP address or updates the IP address with the new domain name.

In the proposed network schema an access mechanism is provided for accessing material via the Web Server. In a typical Web Server, some pages include hyperlinks for accessing other associated pages stored at the same or at different Web servers. In our configuration, some of these hyperlinks comprise links to one or more directories, where directories store URLs for accessing particular Web pages. The access mechanism includes access logic to user by a client/server system via a Web interface with a hyperlink comprising a link to one of the directories, for retrieving from associated directory one or more of said stored URLs and for accessing at least one of said particular Web pages using said retrieved URL.

Directories are repositories of objects which are organized to support locating of those objects. Objects comprise one or more attribute-value pairs. This indirect access to Web pages via hyperlinks to directories has significant advantages for Web page organization and facilitates more flexible methods of Web page access than the known use of hyperlinks which include URLs pointing directly to the target Web pages.

Moreover, an email program (as opposed to web-based email), it probably supports LDAP. Most LDAP clients can only read from a server. Search abilities of clients (as seen in email programs) vary widely. In a typical model when someone write or update information, LDAP does not include security or encryption, so updates usually require additional protection such as an encrypted SSL connection to the LDAP server.

Finally, the NMS could be a graphical user interface (GUI) running on a powerful computer such as a workstation. The server manager polls the servers every 60 seconds to determine if the servers are still running or if they have stopped. The Simple Network Management Protocol (SNMP) is used. The traps could contain information such as setting the server status to up when the server successfully establishes a TCP link with the server manager, setting the server status to down when the TCP link between the server and the server manager is dropped and setting the server status to failed login when the server successfully establishes a TCP link with the server manager, but tried an invalid login.

4 Authentication method

According to the network schema (Fig. 3) a user can be authenticated following the next steps. The first step consists of the requests of a client for an IP address from the DHCP server on the network. Then, the DHCP server dynamically assigns an IP address to the client before it has been authenticated. In the second step the client issues a registration request with the LDAP server and communicates its username, password and the IP address it just obtained from the DHCP server to the LDAP server.

In our approach, the method of communication used by a client is the Hyper Text Transfer Protocol (HTTP), but alternative methods could be used. In this network schema the client does not provide itself with a username and password. The LDAP server [3,4] authenticates the username, password and IP address through an LDAP request to the associated LDAP database. Then, the LDAP request searches the LDAP database for the username, password [8] and the possible IP addresses that the DHCP server could assign. The LDAP database is organized in a tree hierarchy: the root of an Internet address is at the top and the common name associated with the user is at the bottom. A directory lookup process involves stepping through a hierarchy of directory objects in accordance with the particular client request. For example, at Fig. 4 , a named directory object specified in the request Distinguished Name ‘gr.corp.printers’ is located by stepping down the directory hierarchy from the root to ‘gr’, then to ‘corp’ and finally to ‘printers’.

The LDAP database is accessible through an open, standards based protocol such as TCP. If the information is found in the LDAP database, it notifies the LDAP server that the user credentials were verified by returning the authenticated credentials. The LDAP server then sends the authenticated credentials to be stored in the central database. In the described embodiment, the LDAP server communicates with the server manager over a single channel to store the credentials in the central database. Alternatively, the LDAP server could also communicate directly or through some other device with the central database in order to store the authenticated credentials. All updates require additional protection such as an encrypted SSL connection to the LDAP server.

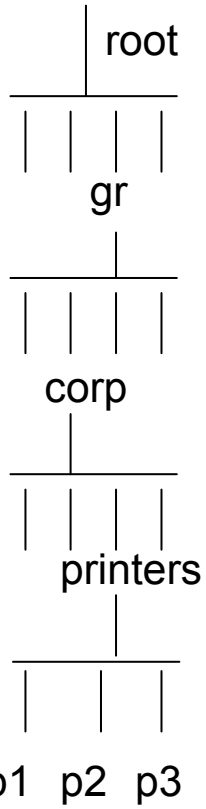

Fig.4. LDAP lookup

5 Conclusion

Lightweight Directory Access Protocol (LDAP) is a fast growing technology for accessing common directory information. LDAP has been embraced and implemented in most network-oriented middleware. As an open, vendor-neutral standard, LDAP provides an extendable architecture for centralized storage and management of information that needs to be available for today's distributed systems and services.

As LDAP matures to a de facto standard, it will eventually replace proprietary directory services in vendor products and other standardized middleware solutions, such as the Distributed Computing Environment (DCE). DCE makes heavy use of a directory service and currently uses its own, specific implementation, called Cell Directory Service (CDS). Moreover, graphical management tools can be added, or existing GUIs may be improved that allow easy configuration and contents management.

As the present model authenticates the actual username, password and IP address, applications such as Voice/Fax over IP and Video Conferencing can benefit because routing and bandwidth considerations are based on source and destination addresses. Therefore, deciding which users can access the network services requires authenticated addresses.

The proposed network model provides authentication using the Lightweight Directory Access Protocol (LDAP). A server manager, which is coupled between the central database and all other servers, communicates configuration information between the central database and each one server in single communication channels. Thus, the model provides trusted information between a central database and all other servers.

References:

[1] RFC2307 An approach for Using LDAP as a Network Information Service, March 1998

[2] RFC3377 Lightweight Directory Access Protocol (v3): Technical Specification, September 2002

[3] RFC2829 Authentication Methods for LDAP, M. Wahl, R. Morgan, May 2001

[4] RFC2222 Simple Authentication and Security Layer, J. Myers, October 1997.

[5] RFC2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, J. Hodges, M. Wahl, May 2000.

[6] RFC2251 Lightweight Directory Access Protocol (v3) Wahl, M., Howes, T., Kille, S., August 1997.

[7] RFC2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions, Wahl, M., Coulbeck, A., Howes, T., Kille, S., December 1997.

[8] RFC3062 LDAP Password Modify Extended Operation, K. Zeilenga , February 2001.