SCIS 2006 The 2006 Symposium on Cryptography and Information Security
Hiroshima, Japan, Jan. 17-20, 2006
The Institute of Electronics, Information and Communication Engineers

Design and Implementation for Integrated Service Authentication System on the basis of Secure LDAP Server

李明坤* 楊中皇†
Ming-Kuen Lee Chung-Huang Yang

Abstract—Authenticating users is one of the more visible aspects of computer security. It is common for everyone owns more than one username. However, remembering and managing all usernames or passwords are both tough task. The usual way of using LDAP for user authentication is to locate the account entry in the directory and then try to bind to the directory as that account, presenting the password or other credentials as received from the user. Setup the LDAP server as the authentication server, it can provide many functions of authentication and management. In this research, we developed an integrated service authentication program to offer username managed and authentication modules put into Unix-like servers to reduce the company's software cost and simplify username managed.

Keywords: Authentication, LDAP, PAM, Security

1 Introduction

The complicated network environment makes every organization need a large number of directories to identify each user in operating systems, application programs and network service. With the organization growing, it becomes very difficult to effectively control, manage and access. If managed improperly, the worst thing could result in the cost to overspend, productivity dropped, and information incompatible. The network authentication requires each user to have one username and password for discerning user's identity. The user must apply for the different usernames to enroll the machine of every kind of authentication requested. It is common for people have two or three usernames, just like people usually have more than one credit card. But remembering each username and password is not an easy thing. Suppose that a system can store all people's username and password, in other words, as long as the user remembers one username, he can enroll and use other services via this systematic authentication. The mechanism like a piece of central agent is responsible for the management of user's information. It is very convenient for this to turn into.

This research proposes LDAP and PAM to solve above mentioned problems. LDAP is the Internet protocol for directory lookups. It is a standard way for applications to request and manage directory information. OpenLDAP is a complete open source suite of client and server applications derived from University of Michigan [6]. The main reason for using OpenLDAP is its open source code. It can support many kinds of authentication ways, such as certificate-based, Kerberos, password, digest, smart card, TLS/SSL and a lot of protected functions [2][5]. It is enough to become a secure authentication server. PAM, called Linux-PAM (Pluggable Authentication Modules for Linux) can insert the pluggable authentication module to support sign-on operations to different authentication domains and the use of multiple mechanisms. PAM is a shared library that makes the administrator switch over freely and select the distinct authentication modules to validate the user. Utilize PAM to incorporate individual server with the question of authentication.

2 Related Work

2.1 LDAP

LDAP (Lightweight Directory Access Protocol) is

* Institute of Information and Computer Education, National Kaohsiung Normal University(NKNU), 116, Ho Ping First Road, Kaohsiung 802, TAIWAN, (captain@icemail.nknu.edu.tw)
† NKNU(chyang@computer.org)

defined in RFC 1777 and originally designed by the University of Michigan, derived from directory access protocol X.500. Though both of those use the same data structure, it is different between LDAP and X.500 essentially. For example, LDAP run directly over TCP/IP, and X.500 builds on OSI stack. LDAP is simpler than X.500 to the code of the things. It is easier than X.500 for LDAP model structure to realize, X.500 structure seems huge and verbose [9]. The latest edition of LDAP at present is the third version. The data model of LDAP is centered around entries, which are composed of attributes. Each attribute has a type and one or more distinct values. The entry can represent a user, a computer, even a department. Each entry has a fully qualified name, the Distinguished Name (DN) defined in RFC 1779. Each component of the DN is called a Relative Distinguished Name (RDN). The DN for any entry is constructed by concatenating the RDN of the entry's ancestors. Entries are organized in a Directory Information Tree (DIT) [3]. The data inside the LDAP server is stored with LDIF (LDAP Data Interchange Format) form that appeared a kind of pure text. This form makes area with line, receive attribute of entries after colon, for example:

dn: cn=david,ou=mis,dc=abc,dc=com
cn: david
objectClass: username
objectClass: posixUsername
objectClass: top
....

This kind of form is easy to pass on, back up and do a large amount of entries modified between the LDAP servers. LDAP can be applied to mail service, authentication system and resource controlled.

2.2 PAM Modules

PAM, pluggable authentication modules on Linux, is a suite of shared libraries for the administrator can replace freely and choose which authentication module to check users. Don't need to rebuild this program when installing. The biggest advantage is its flexibility and extensibility. PAM offered varied authentication methods from pam_permit to the smart card. Depending on your demand, to switch over the authentication mechanisms at any time.

PAM can be considered as a connected interface between "the application program to request authentication service" and "the authentication mechanism of offering authentication service". As to the whole authentication system, PAM is the front-end of application program interface (API) of the authentication system, and the authentication mechanism is the back-end, such as Fig. 1.

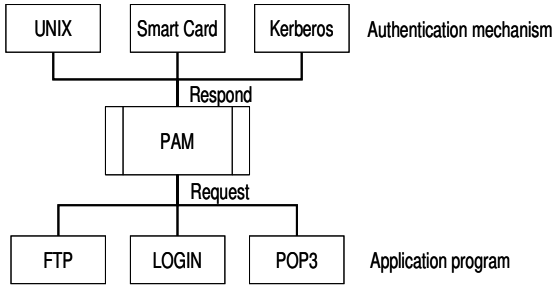

Figure 1: PAM Interface

Because PAM is open standard, most Unix-like systems support PAM. For various kinds of network services, the vendor developed one's own relevant PAM by oneself. PAM has four kinds of authentication types: auth, account, session and password respectively, represent different types of authentication procedures. A particularly powerful feature is that PAM can be stacked, allowing for a choice of authentication methods and perhaps several different policy modules to be active as well [1][7].

2.3 NSS Modules

In order to get necessary information, NSS (Name Service Switch) can be appointed the order of Files or directories to be inquired by the administrator. NSS is a facility found in Linux which allows new name-resolution code to be installed as shared objects without having to rebuild existing libraries [1]. NSS module can be considered as the name converter, it can set the name resolved by the LDAP server at first. For example, if you want to query some information of hosts. You can go copies of machine to inquire /etc/hosts, or turn to DNS. In this research, we use the LDAP server to resolve and analyze information where come from. If you want to make the machine use LDAP for password and group lookups first, set groups file to establish as follows:

```
passwd:     ldap   files
shadow:     ldap   files
group:      ldap   files
```

The effect of this is that lookups will use LDAP first, and then progress to local files (mean by /etc/hosts) for anything not found.

3 Implementation

3.1 System architecture

This research presents an overview of the architecture shows as Fig 2.

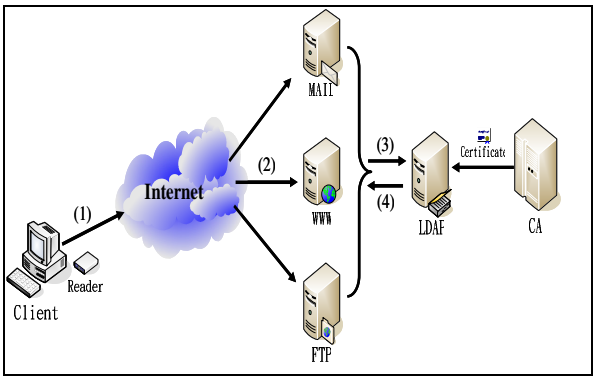

Figure 2: System architecture

The main operating mode is client and server. It is composed of two parts, one is the server accepting the request. We choose OpenLDAP. Setup the LDAP server as the authentication server, and OpenCA is the certificate managed server. We can purchase the certificate to the known enterprises, such as Entrust, Versign etc.., or issue ourselves. This is easy to set up using OpenSSL:

% openssl genrsa –out ldap.key 1024
% openssl req –new –key ldap.key –out ldap.csr

It generates the LDAP server's private key and certificate, and the certificate is signed by CA. LDAP TLS/SSL connections use port 636. This is to say that the communication protocol to make into ldaps:///URL and offers encrypted function to protect the password on transport-layer. The certificate approved by CA can pass to the LDAP server, and the user can download at any time. On the other side, one is client end. The user inputs username and password to access resources, or uses the smart card by inserting the reader to acquire the information stored in the card. The user submits the request of serving to the application server such as WU-FTP or Apache. It will be turned to LDAP looks up user's information via NSS modules. The validity of the request can use PAM modules to prove. PAM, according to the different authentication configure, calls out the shared libraries to carry on verification. Once proving successfully, passing back a value of PAM _ SUCCESS, otherwise passing back a value of PAM_FAILURE. After authentication succeeded, the application program will talk to the user can succeeds in enrolling and using resources of the machine. The flow chart of the whole authentication is such as Fig. 3.

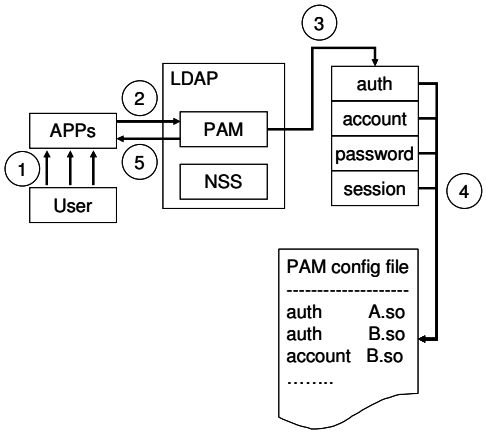

Figure 3: Flow of authentication

3.2 System Implementation

In this research, we use the Mozilla LDAP C SDK to develop LDAP-Client functions and Borland C ++ Builder 6.0 to design an easy and visual pattern of combining authentication with enrollment (see Fig. 4).

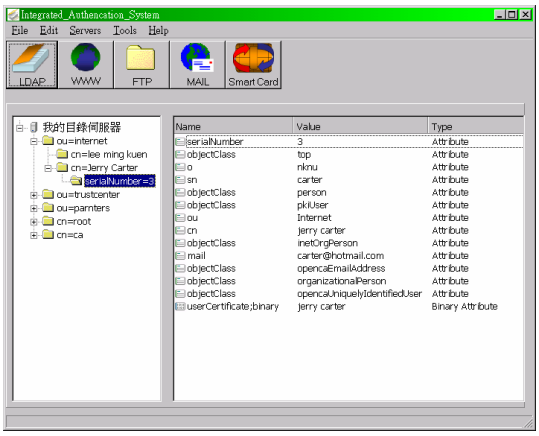

Figure 4: Integrated service auth. Client-side

It not only can connect the LDAP server and query about user's information through this pattern, but also offer the single authentication interface of different network services [4].

For example, a simple LDAP search involves a sequence of 4 operations: ldap _ init, ldap _ bind, ldap _ search and ldap _ unbind. ldap _ init is a initialized LDAP function in charge of starting a session. ldap _ bind is responsible for client authentication. The bind operation allows a client to identify itself to the directory server by using DN and other authentication credentials (the smart card or other information). LDAP supports a variety of authentication methods. ldap _ search is a searched function according to the filter. ldap _ unbind releases the memories to finish the line.(Fig 5)

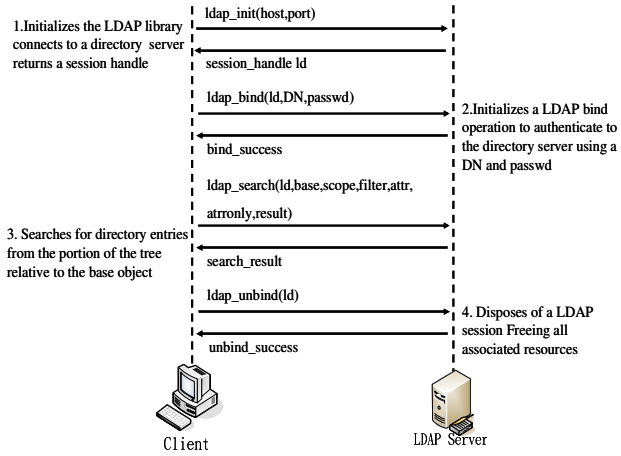

Figure 5: Sequence of a search step

The enroll step is to input username and password, and then submit the serving request through the authentication process. Because the namespace of LDAP can guarantee that each user's name has independent character. For protecting one's password, it can also use the smart card stored a PKC and private key to enroll because it is difficult to duplicate the smart card. In order to simplify the trouble that different servers must input the username again, LDAP can support to assign a group of users can access the specific computer. It is the purpose of using the single identity to access multiple services as well as SSO (Single sign on), and diminishes the distress of remembering varied usernames. It is to make the way to set up a LDIF entry in LDAP, for example:

```
dn: uid=bill,ou=mis,dc=abc,dc=com
uid: bill
cn: bill wang
objectClass: username
objectClass: posixUsername
objectClass: shadowUsername
loginShell: /bin/bash
uidNumber: 680
gidNumber: 200
homeDirectory: /home/bill
userPassword: {crypt}HoYmlMd5giA
host: ldap.abc.com
host: ftp.abc.com
host: www.abc.com
```

This entry is set up inside the LDAP server, and the user can enroll into LDAP, FTP and WWW servers to access resources with "bill" username.

3.3 Security

As to concentrate all entries preserved many sensitive data on the same directory, it becomes very important of security. Some entries should avoid unauthorized users to read. Though all entries are hashed through hash function, but it may be attacked and analyzed by the dictionary attack, or using relay attack to fetch the privilege. Fortunately, OpenLDAP can prevent these things from happening because it has most of the tools needed to do it.

ACL (Access Control List) can control over who may read or who may revise, and assign each user to access resources effectively. Let unconcerned personnel have no chance to contact to the sensitive data. LDAP offers a set of grammars to protect the access of different entries. This is very important inside LDAP directory to store the password, as follows:

```
access to attr=userpassword
        by self write
        by anonymous auth
by * none
```

It allows authorized users to change their own password, and limits other userpassword only to authenticate. However, unauthorized user is unable to revise other people's password. The server will compare with password input by the user and data stored in it. In the other words, it has never sent away the server in password. You needn't be afraid that your password is picked and fetched by the other person.

LDAP also offers two useful mechanisms to protect the password. The first is SASL (Simple Authentication and Security Layer) defined in RFC 2222 can offer both authentication and authorized functions in API and modules. SASL module can support a lot of authentication mechanisms, for instance, Kerberos IV/V, CRAM and DIGEST MD5. The second, LDAP also supports encryption and authentication using TLS/SSL. It can prove the true or false server and protect the password transmitted in the network with a suitable X.509 certificate [8].

4 Conclusion

At present, in Linux servers, such as Telnet, Postfix, WU-FTP, Apache etc…, all can do authentication works through PAM modules. Besides, Squid server also supported, it was used to limit the privilege to use resources on the network.

Setup the LDAP server as the authentication server, it is firmer than the cleartext password, and there is a greater extended space. In this research, we developed an integrated service authentication program to offer username managed and authentication modules put

into Unix-like servers to reduce the company's software cost and simplify username managed. In the future, it will be more extensive for LDAP in merging the network services to enjoy more convenient resources.

References

[1] Andrew Findlay, "Security with LDAP", http://www.skills-1st.co.uk/papers/security-with-ldap-jan-2002/security-with-ldap.singlesided.pdf, 2002.

[2] Brad Marshell, "System Authentication using LDAP", http://quark.humbug.org.au/publications/ldap/system_auth/sage-au/system_auth.html.

[3] G.Carter, *LDAP System Administration*, O'Reilly, 2003.

[4] M.Smith and T.Howes, *LDAP: Programming Directory-Enabled Apps*, sams, 1997.

[5] M. Wahl et al., "Authentication Methods for LDAP", IETF RFC 2829, http://www.ietf.org/rfc/rfc2829.txt, 2000.

[6] OpenLDAP, http://www.openldap.org/

[7] PADL's pam_ldap, http://www.padl.com/OSS/pam_ldap.html

[8] W.Stallings, *Cryptography and Network Security: Principles and Practices, e3*, Person Education, 2003.

[9] V.Koutsonikola and A.Vakali, "LDAP: Framework, Practices, and Trends", IEEE Computer Society, Sep 2004.