IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 2, Ser. III (Mar - Apr 2019), PP 59-70
www.iosrjournals.org

A Unified Identity Management Service Structure Based on LDAP and Kerberos for Organization with .Mail.bd Domain

Jahidul Arafat, Md. Abdul Malek Chowdury, Mysha Nishat Bidhu, Tazkia Binty Faruqui

Department of Computer Science & Engineering (CSE),
Bangladesh Army International University of Science & Technology (BAIUST)
Corresponding Author: Jahidul Arafat

Abstract: *With the development of Internet technology, the increasing of the application service makes the enterprise network management and security more and more complex. Based on it, the military service portals need a system with high performance for identity authentication management. In this paper, a unified identify authentication system through LDAP and Kerberos have been implemented and designed a rational structure of the directory tree. This proposed model also improves the availability and scalability by exploiting directory replication and referral services. In a word where security is a key factor in establishing a reliable IT infrastructure, this paper enables to gain the skills to enable the LDAP directory server along with Kerberos server and client applications to securely communicate via TLS (transport Layer Security). Thus, all the application servers use the user information in the same directory. By this manner, the problem of disagreement and larger maintenance burden can be solved and the system security can be improved across .mil.bd domain.*

Keywords: *Security, LDAP, Kerberos, Identity authentication, Authorization, Directory tree, Single Sign On.*

Date of Submission: 03-04-2019 Date of acceptance: 18-04-2019

I. Introduction

In order to reduce the times of user logging in the independent application system, academic researchers have paid more attention to the unified authentication system since long ago. Such as IBMs Tivoli Access Manager, Microsoft 550 integration system in the windows 2000, and so on. Some of the representative are Kerberos Protocol, Liberty Protocol, Suns Identity Server and Microsoft NET Passport, which support the unified authentication system. [1].With the continuous development of information construction, many domestic organizations and institutions start to consider the problem of integration of application system. Unified and authentication have the focus of the study. Now, the domestic technology of the unified identity authentication is mainly including two parts: one is the digital signature authentication based on certificate [2]. But most of them maintain in solving the individual demand level. Perfection, unified holistic solution or standard are not formed. Another is based on username and password like UNIX, which is vulnerable to attack and low security. This paper researches the identity management service structure system based on LDAP and Kerberos, according to application requirement of portal system of Bangladesh Army in the .mil.bd domain having many sponsored and affiliated organizations including Bangladesh Army International University of Science and Technology in the edu.bd domain. The system concentrate service information on directory service, which allows different place and system client to access easily. By rational use of directory service, the system can effectively reduce data repeatability and the burden of the management work. This paper thereby presents a user modeling server that is based on a very different type of data repository, namely directory systems. The use of directories for user modeling systems has already been considered, and dismissed, by [3]. This paper however argues that the reasons for their rejection do not hold true anymore today. Rather, directory systems offer a number of significant advantages over database systems that should make them the data storage of choice for user modeling servers in the currently prevailing application scenarios for user modeling. In Section 2 of this paper, a brief review of knowledge and different database and directory management systems are given that have served as data repositories of various centralized and decentralized user modeling systems while specifically focusing on the Lightweight Directory Access Protocol (LDAP). In Section 3, the problems addressed by this study are discussed. Section 4 presents the architecture of Unified Identity Management System, an LDAP and Kerberos -based user modeling server that was used successfully in two different application domains, one for base army .mil.bd domain and one for its sponsored and affiliated organizations domain. The proposed architecture also includes solutions for scalability (to a user population size that is realistic for contemporary web applications), and for user modeling with intermittent network connectivity such as in user-adaptive mobile systems. In Section 5, a simulation experiment to test the performance and scalability of the user modeling server are described in real world application scenarios. A theory of how such experiments should be conducted, based on available client-side usage data (and not server logs that have been exclusively used so far) are proposed Section 6, finally, summarizes the contributions and the resulting conclusions.

II. Literature Review

This section gives an overview of how user models have been stored in major generic user modeling systems that have 2 been developed to date. GUMS [4], [5] allowed programmers of user-adaptive systems, the definition of simple stereotype hierarchies and, for each stereotype, of Prolog clauses describing stereotype members and rules prescribing the systems reasoning about them. BGP-MS [[1],[6],[7][3]] allowed assumptions about the user and stereotypical assumptions about user groups to be represented in a first-order predicate logic. PROTUM [3] was implemented in IF Prolog and represented user model content as a list of constants, each with associated type (i.e., observed, derived from stereotype, default) and confidence factor. UMT [5] allowed the user model developer the definition of hierarchically ordered user stereotypes, and of rules for user model inferences and contradiction detection. TAGUS [[8], [9]] represented assumptions about the user in first-order formulas, with meta-operators expressing the assumption types. It allowed for the definition of a stereotype hierarchy, a library of misconceptions, and a number of user models. um[4] was a user modeling toolkit that represented assumptions about users knowledge, beliefs, preferences and other characteristics as attribute-value pairs. DOPPELGNGER [4], [5], [10] was a user modeling server that accepted information about users from hardware and software sensors, collected them in user models that were stored on the server, and allowed learning algorithms to operate upon the sensor data and upon user models. Group Lens [11] originally employed various collaborative filtering algorithms [[3], [12]] for predicting users interests, based on explicitly provided users ratings, implicit ratings derived from users navigation, and transaction histories (e.g., shopping basket operations, purchases). Group Lens stored all user ratings in a database, but kept a correlation matrix of all ratings in cache memory during runtime. With the exception of Group Lens and, recently, Personis [4] and PersonisLite [8], none of the developed generic user modeling systems seemingly paid much attention to appropriate storage mechanisms. In the simplest case, all user models were read from secondary storage at launch time (or were already part of the program code of the user modeling system in the first place). In the most sophisticated case, the model of the current user was individually read from a file at the beginning of the session with the respective user, and saved to a file thereafter. During a session with a user, the complete user model was maintained in main memory. In many cases, user models were also tightly intertwined with the programming language (e.g., part of the LISP or PROLOG space). It is obvious that this approach does not scale up when the number of users increases. This disregard of storage considerations is not surprising though since these generic systems were either designed for single-user applications only, or never tested with a larger number of parallel applications and users. Researchers regarded other properties of generic user modeling systems as more important, such as generality including domain independence, representational expressiveness, and inferential capabilities (see [10]).

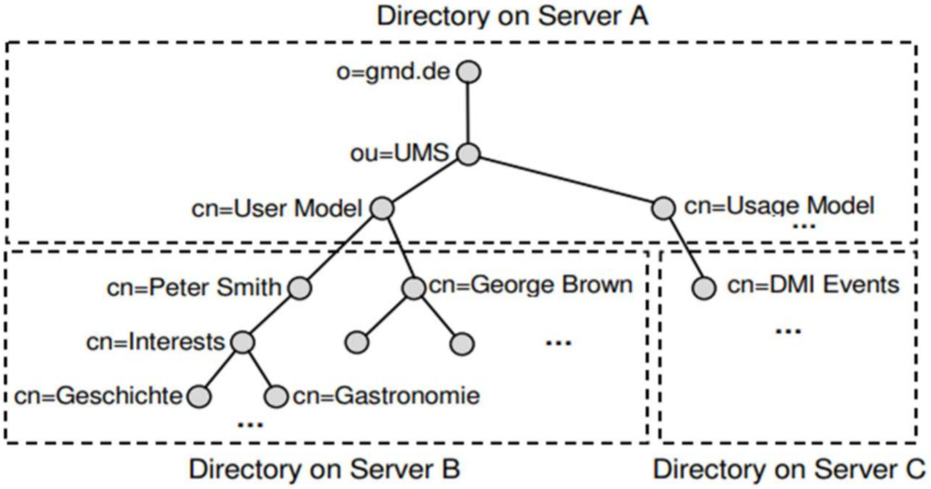

Fig. 1. Distributed directory (based on [2]).

Directories on the other hand are specialized database management systems that were originally designed for maintaining information about people in organizations, and later extended to also include information about devices and services on a network. They are crafted to meet the needs of a wide range of applications and are based on international standards that guarantee interoperability between implementations of different developers and vendors. The first of these international standards promulgated by ITU-T and ISO in the late 1980s was the Directory Access Protocol (DAP, [13]). DAP was intended to be used by clients for accessing an X.500 directory service. It did not gain much popularity, mainly because it was too complex to be implemented and deployed on the hardware that was typical for that time. During the following years, LDAP emerged from the X.500 protocol family as a lightweighted alternative.

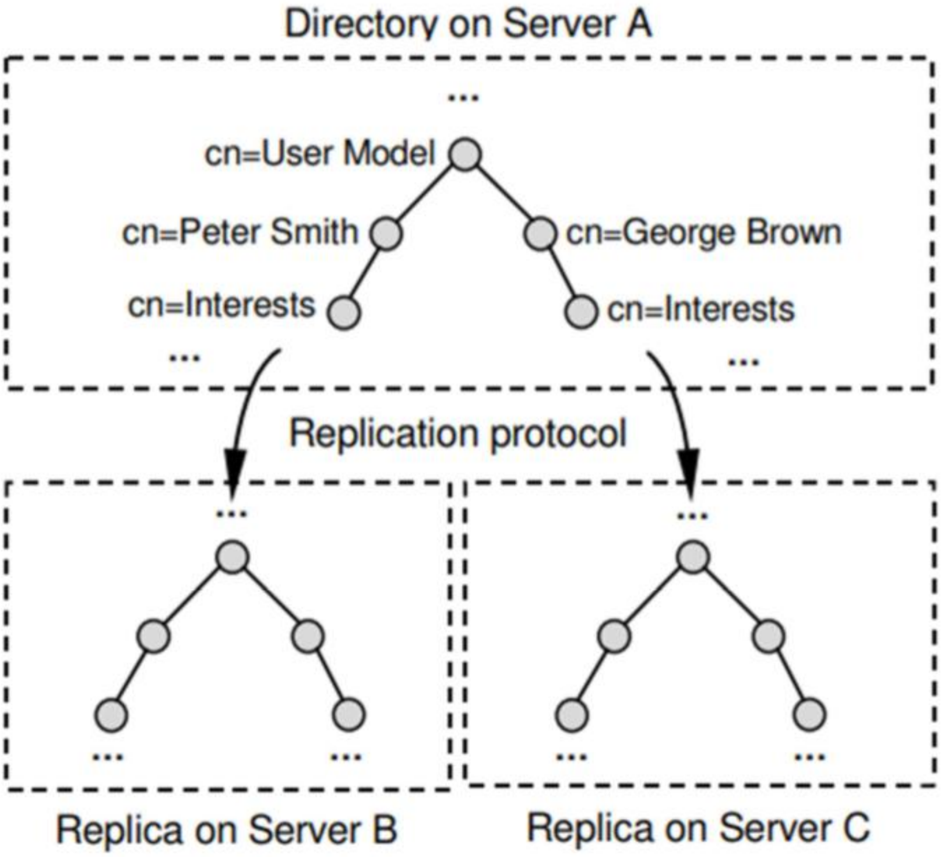

Fig. 2. Replicated User Model (based on [Howes et al. 1999]).

LDAP removed excessive complexity from X.500 DAP, significantly reduced resource requirements, and took advantage of the popular TCP/IP rather than the OSI protocol stack. At the same time LDAP still preserved many strengths of X.500, including its information model (see [Fink 2004]), its versatility, and its openness. Many commercial systems have been developed that are largely LDAP-compliant, including network-wide address books (e.g., Lotus Notes[4] and Microsoft Exchange [14]), network operating system directories (e.g., Microsoft Active Directory [5], Novell eDirectory [3], IBM Tivoli Directory Server [13], and Sun Java System Directory Server [7]), and specialpurpose Internet directories (e.g., [7], [3]). In the following, LDAP by means of the following four models will be briefly introduced: (a) Information model, which defines the types of data that can be stored in a directory. (b) Naming model, which describes how to organize and refer to directory data. (c) Functional model, which prescribes how to access directory data. (d) Security model, which defines how to control access to directory data. Moreover, a distributed directory is in many cases cheaper to implement and simpler to manage (see [2] for more information on distributed LDAP deployments and resulting advantages). Replication of user model information is very attractive for remote user-adaptive applications. It allows these applications to temporarily duplicate the whole user model or substructures, to manage them locally while offline (including local edits and additions), and to synchronize the local copies with the original data (which may have also changed in the meantime) when this becomes possible again. Figure 2 shows a situation where the User Model on server A has been replicated in its entirety both on servers B and C (we omitted the labels in the replicas for reasons of brevity). Performance and scalability are very important criteria for user modeling applications, particularly when those are to be deployed to the web. Directories have been designed to meet the needs of a wide variety of Internet and intranet applications (e.g., e-mail servers and clients, Web server applications and browsers, groupware servers and clients, and lightweight database applications). Scalability is therefore of paramount importance for directories since the number of applications that will utilize them is often not known at the time of deployment. Databases, in contrast, are often designed for a dedicated set of database applications. Adherence to open standards in the design of user modeling clients and servers is very important since this improves their interoperability. There already exist efforts in some subfields of user modeling to come up with standards (e.g., [4-8]), although without many results so far. [14] LDAP is highly standardized. The most important areas of standardization are: (a) LDAP protocol specifications (b) Proposed extensions to LDAP version 3 (c) Related Internet standards or proposed standards (d) Additional security standards besides SASL (e) The LDAP Data Interchange Format (LDIF, RFC 2849) and (f) A C programming interface for LDAP (RFC 1823), and several APIs for Java that are available as Internet Drafts.

III. Problem Addressed by This Study

LDAP server is the core of database of the unified identify authentication system based on directory service id proposed in this model. The system storing user identify, role and access control information, establishes corresponding strategy service system. In addition, portal application system as a whole is managed and authorization service of policy made by administrator is executed . So, the proposed model is designed to solve the following several key problems:

A. Build Directory Information Base

Base DN (Distinguished Name) is the foundation of the whole unified user authentication platform. A user or organization information is collected and stored with hierarchical structure way for unified management, which assure consistency and integrity of the data, and, server all kinds of application system of the enterprise including enterprise portal, ERP system, yellow pages, network billing system, and so on. First, the DN can store the information of all kinds of the object in the enterprise net system, including network devices, all kinds of server, sector organization, user and so on. The storage of the information is not only concentrated but also distributed in different geographic position(or different network environment). Secondly, The DN adapts to all kinds of new needs, makes appropriate adjustments, flexible changes or expands the information in the database if it is necessary, but, does not produce great influence for the existing data as a result of it’s good expansibility and applicability. Thirdly, The DN provides standard directory access to all kinds of application based on LDAP caused of general LDAP interface[13].

B. Provide Information Service

The application program used for viewing or modifying the information in the directory information database was achieved. The application program must have the following some function: access based on Web, queried and modified information through Web, providing the ability of distribution access and management, offering different levels of information and safely be transferred information between the LDAP client and be severed with the authentication and authorization.

C. Unified Identity Authentication

The application program authenticating with the directory server was achieved, which was used for unifying the authentication function of the different application servers. Users identify and access management in the whole portal system were done through building an independent, high safety and reliability identify authentication and user permission management system. Than it changes the tradition and isolation of the identify authentication of each application system, and offers a possibility to realize a higher level of service in the intranet[5].

IV. Proposed System Architecture of Unified Identity Management on .Mail.bd Domain

Bangladesh Army and all of its sponsored educational organizations composed the different security domain through the authentication systems which are independent of one another. After the user authenticated successfully in his security domain, the users can access all parts of the portal and the different application systems with Single Sign On (SSO). In addition, the user of Bangladesh Army can access the specified portal and application system of all province companies with cross-domain SSO, which must have the aid of the cascading authentication among the identity servers.

The structure includes three parts:

A. Directory Server

The directory system of Bangladesh Army was made up of 2-tier architecture, which were the Army main directory and each sponsored organizations directory. According to the condition of network infrastructure, the requirement of personnel management and the requirement of application system deployment, each sponsored organization selectively established the geographical level of directory (That is third-tier deployment of the directory). In order to promote the construction of directory system and implement the concatenation of levels of directory systems in the whole net, various directories of Bangladesh Army, Sponsored and affiliated organization and geo-location must be designed in strict accordance with the unified directory tree and the directory schema. This 2- tier directory adapt to the geographical distribution of the Bangladesh Army and its sponsored and affiliated organizations, reduce network bandwidth use, increase query efficiency of the directory, and conform to design standard of the super large enterprise distributed directory system at present. In logic, the proposed identity directory in the whole net has the most completely user identity information and server user identity authentication and storage inside of main grid of Bangladesh Army. According to the organization structure, the design of directory tree can select a flat structure, showed as follows: The proposed identity directory of sponsored and affiliated organizations has the most completely user identity information and server the user identity authentication and storage inside of each organization. From the perspective of directory synchronization, the design of identity directory tree selected a flat structure, showed as follows:

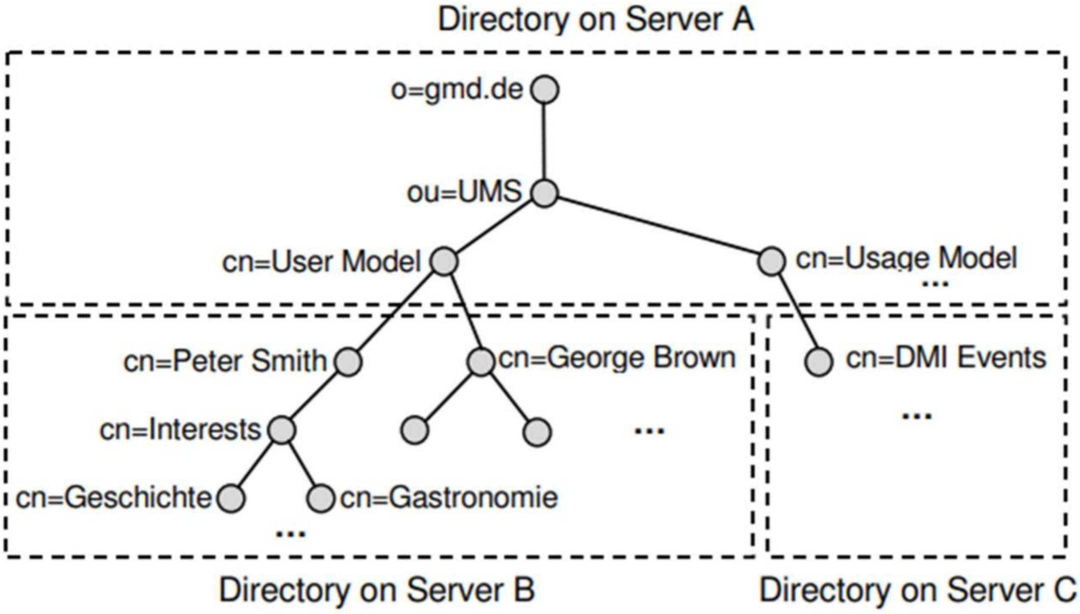

Fig. 3. Proposed Directory tree of the whole Bangladesh Army Net])

B. Identity Management

Identity management synchronized user information in the authoritative data source (such as human resource system) to identity directory is through identity synchronization tool. And, it can sent user information to each application system, according to a pre-specified strategy. That achieved Account to automatically creating, changing, canceling, replaced the existing manual account management. The proposed unified system provides an Integrated Identity Manager (IIM) to synchronize the identity directory with database, directory and standard application. And, it stored the log information of key events in the database for later audit.

C. Authentication System

The proposed authentication system made up by access gateway and identity authentication management server, is the entry of unified access of portal in the Bangladesh Army or sponsored or affiliated organizations and most application system, provides centralized authentication of user identity and security access of portal and application system. Using the mechanism of identity injection, access gateway based on reverse proxy realizes the SSO of portal and application system using Kerberos 5. Most people will not use Kerberos by itself; once an user is authenticated (Kerberos), we figured out what this user can do (authorization). And that would be the job of programs such as LDAP.

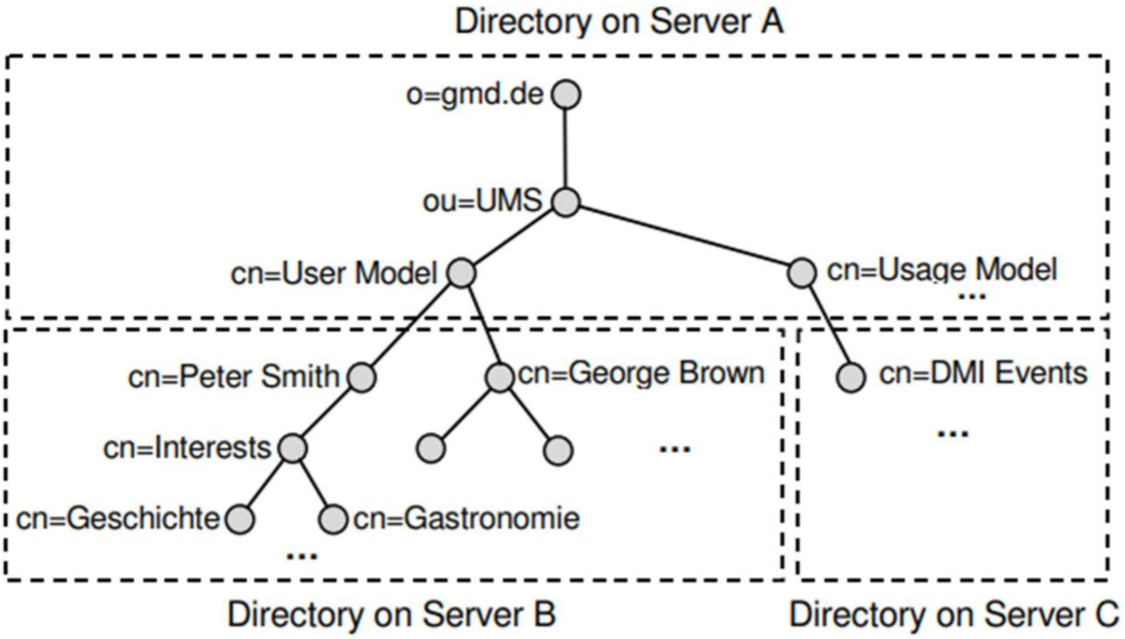

Fig. 4. Proposed Directory tree of the sponsored and affiliated organizations of Bangladesh Army]).

Replicating a Kerberos principal database between two servers can be complicated, and adds an additional user database to the network. Fortunately, MIT Kerberos can be configured to use an LDAP directory as a principal database.

V. Experimental Analysis and Result

A. Server Installation Requirements

Directory of state grid corporation, authentication and identity management need three computers at least. Considering of high availability reasons, computers increased to eight HA or cluster architecture at least to avoid abnormal authentication of application system when single point of failure of wrong. Installation of 3 computers is shown as follows:

Table no 1 : Authentication Server Requirements

CPU	Two 3Ghz CPU or higher
RAM	4.0 GB 533Mhz ECC DDR2
Hard Disk	100GB
Network	One integrated controller of 1000Mb/s Ethernet
Operating System	Red Hat Enterprise Linux Server 7.5

Table no 2 : Gateway Server Requirements

CPU	Two 3Ghz CPU or higher
RAM	4.0 GB 533Mhz ECC DDR2
Hard Disk	100GB
Network	Two integrated controller of 1000Mb/s Ethernet
Operating System	Red Hat Enterprise Linux Server 7.5

Table no 2 : IIM Server Requirements

CPU	Two 3Ghz CPU or higher
RAM	4.0 GB 533Mhz ECC DDR2
Hard Disk	200GB
Network	One integrated controller of 1000Mb/s Ethernet
Operating System	Red Hat Enterprise Linux Server 7.5

B. Implementation phase

This paper has used the mil.bd domain and the baiust.mil.bd hostname. This hostname has been resolved by the /etc/hosts file. It can also be resolved by DNS.

1) Stage 01: Setup Procedure

Step 1: Install the following packages:

```
# yum install −y openldap openldap−clients openldap− serversmigrationtools
```

Step 2: Generate a LDAP password from a secret key (here redhat):

```
# slappasswd −s redhat −n > /etc/openldap/passwd
```

Step 3: Generate a X509 certificate valid for 365 days:

```
# openssl req -new -x509 -nodes -out /etc/openldap/certs/cert.pem \
-keyout /etc/openldap/certs/priv.pem -days 365
Generating a 2048 bit RSA private key
.....+++
.............+++
writing new private key to '/etc/openldap/certs/priv.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:baiust.mil.bd
Email Address []:
```

Step 04: Secure the content of the **/etc/openldap/certs** directory:

```
# cd /etc/openldap/certs
# chown ldap:ldap *
# chmod 600 priv.pem
```

Step 05: Prepare the **LDAP** database:

```
# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

Step 06: Generate database files:

```
# slaptest
53d61aab hdb_db_open: database "dc=my-domain,dc=com": db_open(/var/lib/ldap/id2entry.bdb) failed: No such file or directory (2).
53d61aab backend_startup_one (type=hdb, suffix="dc=my-domain,dc=com"): bi_db_open failed! (2)
slap_startup failed (test would succeed using the -u switch)
```

Step 07: Change **LDAP** database ownership

```
# chown ldap:ldap /var/lib/ldap/*
```

Step 08: Activate the **slapd** service at boot:

```
# systemctl enable slapd
```

Step 09: Start the **slapd** service:

```
# systemctl start slapd
```

Step 10: Check the **LDAP** activity:

```
# netstat -lt | grep ldap
tcp        0      0 0.0.0.0:ldap  0.0.0.0:*   LISTEN
tcp6       0      0 [::]:ldap   [::]:*  LISTEN
```

Step 11: To start the configuration of the **LDAP** server, add the **cosine** & **nisLDAP** schemas:

```
# cd /etc/openldap/schema
# ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f cosine.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=cosine,cn=schema,cn=config"
# ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f nis.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=nis,cn=schema,cn=config"
```

Step 12: Then, create the **/etc/openldap/changes.ldif** file and paste the following lines
(replace **PASSWORD** with the previously created password
like **{SSHA}l8A+0c+lRcymtWuIFbbc3EJ1PRZz9mGg**):

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=example,dc=com
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=Manager,dc=example,dc=com
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootPW
olcRootPW: PASSWORD
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/cert.pem
dn: cn=config
changetype: modify
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/certs/priv.pem
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: -1
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by
dn.base="cn=Manager,dc=example,dc=com" read by * none
```

A problem with **olcTLSCertificateFile** and **olcTLSCertificateKeyFile** has been reported in recent versions of **OpenLDAP** coming with **RHEL 7.5**; these attributes have to be modified at the same time:

```
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/cert.pem
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/certs/priv.pem
```

Step 13: Send the new configuration to the **slapd** server:

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f /etc/openldap/changes.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={2}hdb,cn=config"
modifying entry "olcDatabase={2}hdb,cn=config"
modifying entry "olcDatabase={2}hdb,cn=config"
modifying entry "cn=config"
modifying entry "cn=config"
modifying entry "cn=config"
```

Step 14: Create the **/etc/openldap/base.ldif** file and paste the following lines:

```
dn: dc=mil,dc=bd
dc: example
objectClass: top
objectClass: domain
dn: ou=People,dc=mil,dc=bd
ou: People
objectClass: top
objectClass: organizationalUnit
dn: ou=Group,dc=mil,dc=bd
ou: Group
objectClass: top
objectClass: organizationalUnit
```

Step 15: Build the structure of the directory service:

```
# ldapadd -x -w redhat -D cn=Manager,dc=example,dc=com -f /etc/openldap/base.ldif
adding new entry "dc=mil,dc=bd"
adding new entry "ou=People,dc=mil,dc=bd"
adding new entry "ou=Group,dc=mil,dc=bd"
```

Step 16: Create two users for testing:

```
# mkdir /home/guests
# useradd -d /home/guests/ldapuser01 ldapuser01
# passwd ldapuser01
Changing password for user ldapuser01.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
# useradd -d /home/guests/ldapuser02 ldapuser02
# passwd ldapuser02
Changing password for user ldapuser02.
New password:
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
```

2) *Stage 02: User authentication with Kerberos 5*

Configuring a Secondary KDC using the LDAP backend is similar to configuring one using the normal Kerberos database.

Step 01: First, install the necessary packages. In a terminal enter:

```
yum install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

Step 02: Next, edit /etc/krb5.conf to use the LDAP backend:

```
[libdefaults]
        default_realm = MIL.BD

[realms]
        MIL.BD = {
                kdc = armyBase.mil.bd
                kdc = armyOthers.mil.bd
                admin_server = armyBase.mil.bd
                admin_server = armyBase.mil.bd
                                        default_domain = mil.bd
                database_module = openldap_ldapconf
        }
[domain_realm]
        .mil.bd = MIL.BD
...
[dbdefaults]
        ldap_kerberos_container_dn = dc=mil,dc=bd
[dbmodules]
        openldap_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=mil,dc=bd"
                # this object needs to have read rights on
                # the realm container, principal container and realm sub-trees
                ldap_kadmind_dn = "cn=admin,dc=mil,dc=bd"
                # this object needs to have read and write rights on
                # the realm container, principal container and realm sub-trees
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps:// armyBase.mil.bd ldaps:// armyOthers.mil.bd
                ldap_conns_per_server = 5
        }
```

Step 03: Create the stash for the LDAP bind password:

```
sudo kdb5_ldap_util -D  cn=admin,dc=mil,dc=bd stashsrvpw -f \
/etc/krb5kdc/service.keyfile cn=admin,dc=mil,dc=bd
```

Step 04: Finally, start the *krb5-kdc* daemon:

```
sudo systemctl start krb5-kdc.service
```

Verify the two ldap servers (and kerberos by extension) are in sync.

You now have redundant KDCs on your network, and with redundant LDAP servers you should be able to continue to authenticate users if one LDAP server, one Kerberos server, or one LDAP and one Kerberos server become unavailable.

3) Stage 03: User Account Migration

Step 01: Go to the directory for the migration of the user accounts:

```
# cd /usr/share/migrationtools
```

Step 02: Edit the **migrate_common.ph** file and replace in the following lines:

```
$DEFAULT_MAIL_DOMAIN = "mil.bd";
$DEFAULT_BASE = "dc=mil,dc=bd";
```

Step 03: Create the current users in the directory service:

```
# grep ":10[0-9][0-9]" /etc/passwd > passwd
# ./migrate_passwd.pl passwd users.ldif
# ldapadd -x -w redhat -D cn=Manager,dc=mil,dc=com -f users.ldif
adding new entry "uid=ldapuser01,ou=People,dc=mil,dc=bd"
adding new entry "uid=ldapuser02,ou=People,dc=mil,dc=bd"
# grep ":10[0-9][0-9]" /etc/group > group
# ./migrate_group.pl group groups.ldif
# ldapadd -x -w redhat -D cn=Manager,dc=mil,dc=bd -f groups.ldif
adding new entry "cn=ldapuser01,ou=Group,dc=mil,dc=bd"
adding new entry "cn=ldapuser02,ou=Group,dc=mil,dc=bd"
```

Step 04: Test the configuration with the user called **ldapuser01**:

```
# ldapsearch -x cn=ldapuser01 -b dc=mil,dc=com
```

4) Stage 04: Firewall Configuration

Step 01: Add a new service to the firewall (**ldap**: port **tcp 389**):

```
# firewall-cmd --permanent --add-service=ldap
```

Step 02- Reload the firewall configuration:

```
# firewall-cmd --reload
```

Step 03: Edit the **/etc/rsyslog.conf** file and add the following line:

```
local4.* /var/log/ldap.log
```

Step 04: Restart the **rsyslog** service:

```
# systemctl restart rsyslog
```

C. Test Results and Performance Analysis

1) Test Results :

When user access the enterprise portal, username and password will be submitted to directory service system. Directory server will compare them with data in the directory database, legal user can pass the authenticated of LDAP. If user is illegal, the system cannot be logged in successfully.

2) Performance Analysis :

System design implemented centralized management and authorization of user. At the same time, through service of interface provided by user information management module, system served user centralized management of different user management systems. And system simplified integration process through registration function of application system. New application system without its own user system used unified identity authentication system to realize user's authentication and authorization. By way of unified authentication, users account and password are transferred only in the first login and transmitted encrypted. During user periodic authentication, asking message and authentication identifier for each authentication are different, so replay attack could be prevented effectively [8]. System bound user account with IP to timing authenticate to make sure user on line. That prevent certifier IP attack.

VI. Conclusion

In the paper, system use directory tree structure and single sign on application system to solve the problem of user authentication of Bangladesh Armys .mil.bd domain and authorization of accessing portal of province. Using TOKEN of mutual trust between Bangladesh Army and portal of its affiliated and sponsored organizations to realize unified identity authentication, user convenient and shortcut access all authorized application services, at the same time, system security is protected.

References

[1]. Hong Yan, “JAVA and model”,Beijing: Electronic Industry University Press,2002

[2]. BruceSchneier, “Applied Cryptography”, Beijing: Machinery Industry Press,2005:361-367

[3]. Xiang Li; Ai-nong Chao, “Research and application of LDAP in uniform identity authentication”, Journal of Computer Application,2008-S1,pp.28-32

[4]. Cheng-long Zhang, Dong Wang, “Exploration and Practice of Directory Service Based on LDAP”, Financial Computerizing, No144,Apr.2012,pp.81-83

[5]. W. Yeong, T.Howes, S.Kille. Lightweiht Directory Access Protocol.RFC1777, Mar.1995

[6]. Jie Lan, “Research on Single sign-on System”, Technological Development of Enterprise,Vol.31 No.5,Feb.2012,pp.73-74

[7]. Mohammad Salim; M Sana Akhtar; Mohammed A Qadeer,“Data Retrieval and Security using Lightweight Directory Access Protocol”, Proceedings of 2009 Second International Workshop on Knowledge Discovery and Data Mining,Jan.2009

[8]. Wikipedia, “Replay Attack”, http://en.wikipedia.org/ wiki/Replay_attack, Feb.2012.

[9]. Persistent (2006). Persistent. http://www.persistentsys.com.

[10]. Pohl, W. (1998). Logic-Based Representation and Reasoning for User Modeling Shell Systems. Sankt Augustin, Germany, infix.

[11]. Razmerita, L., A. Angehrn and A. Maedche (2003). Ontology-Based User Modeling for Knowledge Management Systems. In: P. Brusilovsky, A. Corbett and F. De Rosis, eds: User Modeling 2003: 9th International Conference, UM 2003. Heidelberg, Germany, Springer Verlag: 213-217. http://springerlink.metapress.com/link.asp?id=thw9rmvmvklx9hac.

[12]. Rozanski, H., G. Bollman and M. Lipman (2001). Seize the Occasion: Usage-based Segmentation for Internet Marketers. http://www.strategy-business.com/media/pdf/03-20-01_eInsight.pdf.

[13]. Schreck, J. (2003). Security and Privacy in User Modeling. Dordrecht, Netherlands, Kluwer Academic Publishers. http://www.security-and-privacy-in-user-modeling.info.

[14]. Shukla, S. and A. Deshpande (2000). Tutorial: LDAP Directory Services – Just Another Database Application? 2000 ACM SIGMOD International Conference on Management of Data, New York, NY. http://www.pspl.co.in/presentation/sigmod2000_directory_database_tutorial.pdf.

[15]. Sun (2006). Sun Java System Directory Server Enterprise Edition. http://www.sun.com/software/products/directory_srvr_ee/.

[16]. Switchboard (2006). Switchboard. http://www.switchboard.com/. Tornago (2006). Net Perceptions. http://www.tornago.com.

[17]. VanderMeer, D., K. Dutta and A. Datta (2000). Enabling Scalable Online Personalization on the Web. 2nd ACM Conference on Electronic Commerce, Minneapolis, MN, ACM, 185-196. DOI: 10.1145/352871.352892. Vassileva,

[18]. J., G. McCalla and J. Greer (2003). "Multi-Agent Multi-User Modeling in I-Help." User Modeling and User-Adapted Interaction: The Journal of Personalization Research 13(1+2): 179-210. DOI: 10.1023/A:1024072706526.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Do ThiBich Ngoc. " A Unified Identity Management Service Structure Based on LDAP and Kerberos for Organization with .Mail.bd Domain" IOSR Journal of Computer Engineering (IOSR-JCE) 21.2 (2019): 59-70.