LDAP Authentication Guide

extreme computing

REFERENCE
86 A2 41FD 00

extreme computing

LDAP Authentication Guide

Software

July 2009

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A2 41FD 00

Trademarks and Acknowledgements

The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Table of Contents

Preface

Scope and Objectives

This guide describes how to install and configure LDAP authentication solution on Bull extreme computing clusters.

Intended Readers

This guide is for Administrators of Bull extreme computing systems.

Important

The Software Release Bulletin contains the latest information for your delivery. This should be read first. Contact your support representative for more information.

Note The Bull Support Web site may be consulted for product information, documentation, downloads, updates and service offers:
http://support.bull.com

Highlighting

- Commands entered by the user are in a frame in 'Courier' font, as shown below:

```
mkdir /var/lib/newdir
```

- System messages displayed on the screen are in 'Courier New' font between 2 dotted lines, as shown below.

```
Enter the number for the path :
```

- Values to be entered in by the user are in 'Courier New', for example:

  `COM1`

- Commands, files, directories and other items whose names are predefined by the system are in '**Bold**', as shown below:

  The **/etc/sysconfig/**dump file.

- The use of *Italics* identifies publications, chapters, sections, figures, and tables that are referenced.

- < > identifies parameters to be supplied by the user, for example:

  `<node_name>`

⚠ WARNING
A Warning notice indicates an action that could cause damage to a program, device, system, or data.

Chapter 1. Introduction

The authentication of the users in an extreme computing cluster is a specific problem, since it must deal with a high security level, the guarantee of consistency, and the adaptation to the number of nodes and users. Using a Directory service based on LDAP (Lightweight Directory Access Protocol) is a pertinent solution, which allows the information related to users to be centralized. This solution is secured (with an encryption mechanism), its design allows to the size of the cluster to be adapted, and it provides redundancy mechanisms based on replication.

The extreme computing clusters use an OpenLDAP server, integrated in the Linux distributions associated with the name service cache daemon (**nscd**), which allows the number of requests towards the server to be reduced. To guarantee an optimum behavior, the OpenLDAP replication mechanisms are used. In addition, an enhanced OpenSSH server provides the SSH public keys for the users through the directory, which means that the keys are no longer managed locally.

Chapter 2. Security Issues

2.1 Using encryption for the communications

To protect sensitive data that go through the network, it is necessary to activate the encryption mechanisms. With LDAP, the TLS (Transport Layer Security) protocol is used, in order to encrypt all the connections and communications between the clients and the server(s).

2.2 Configuring the Access Control Lists (ACL)

Some data of the directory (for example the user passwords) must be protected so that only the authorized user(s) can access this data. It is necessary to guarantee that only the user and the administrator can access the data (as this is done in local with the shadow password).

2.3 Choosing Passwords

At different stages of the LDAP configuration, passwords will be required to guarantee the security. It is necessary to choose passwords carefully using basic rules such as: more than 8 characters, mixing capitals and lowercase letters and special characters, no dictionary word, etc. A trivial password or a password easy to discover with automatic tools cannot guarantee the security of the LDAP authentication solution.

2.4 Certificates

Certificates are used to manage encryption between LDAP clients and servers. These certificates enable the entities that want to communicate to be undeniably proofed. As long as the Cluster operates in a private and controlled network, the certificates can be auto-signed (so do not require the signature of an external authority).

2.5 Broadcasting Public Keys

The public keys, which are used for authentication of SSH connections (password-less) is not sensitive data, since they can be used only with the corresponding private keys. On the other hand the private keys must be securely stored and transmitted. In a Cluster the users generally have a shared account on all the nodes of the cluster (NFS, etc which allows a consistent set of public / private keys in their home directory. The public keys can be stored in the LDAP directory in order to guarantee a global consistency and to enable the users to access all the hosts that are referenced in their set of public keys (equivalent to the **authorized_keys** file).

2.6 Using unique SSH Host Keys

Public / private keys are used by SSH to check the authenticity of a host. By default, all the nodes of the cluster that have been deployed with the same image have the same pairs of host keys. The system does not report an alert because the StrictHostKeyChecking option is set to no. To improve the security, it is recommended to deploy unique pairs of host keys for each node, and not to set the StrictHostKeyChecking option to no.

Chapter 3. Prerequisites

3.1 Time Synchronization

It is very important to ensure the time synchronization within the Cluster. To do this, a system such as Network Time Protocol (NTP) is required on all servers and clients, immediately at installation of LDAP and not only at activation.

3.2 Required packages (dependencies)

In addition to the OpenLDAP packages delivered with bullx cluster suite, the following packages must be installed on all nodes (servers and clients):

- expect
- openldap-clients
- perl-AppConfig
- perl-Crypt-PasswdMD5
- perl-String-ShellQuote
- perl-LDAP
- perl-Net-SSH
- perl-Net-SCP
- perl-threads
- openssl-perl
- perl-Term-ReadKey

3.3 Network Requirements

The communications between the server and the clients must be authorized in both directions. If the LDAP replication is used, it is also necessary to authorize the exchanges between the various servers (Master/Slave) and also towards and from all LDAP clients.

3.4 OpenSSH LPK (LDAP Public Key)

To benefit from the centralization of the public keys for SSH connections (replacement of the **authorized_keys** local file), Bull enhanced packages for OpenSSH server and client are required. These packages are installed by default.

Chapter 4. Configuring the LDAP Servers

Several steps are necessary to install and configure the LDAP servers:

1. Install the required packages
2. Configure the Master Server
3. Configure the Slave Server
4. Configure the LDAP client on the Master Server
5. Generate Certificates for TLS
6. Initialize the Directory
7. Start the ldap-auth service on the Master Server
8. Check Directory Operation on the Master Server
9. Activate the Slave Server

Note It is recommended to use the Management Node for the LDAP Master Server, and the Secondary Management Node for the LDAP Slave Server.

4.1 Install the required packages

Once the pre-requisites packages installed (see 3.2 *Required packages (dependencies)*), you have to install the packages related to authentication by LDAP, as well as associated tools.

1. Insert the **Bull XHPC DVD**, and go to the **BONUS RPMs** directory.

Note In the following commands, the version numbers should be set according to the RPMs present on your DVD.

2. Install the libraries package on all nodes that will supply or ask for the LDAP authentication service (server and clients):

```
# rpm -ivh openldap-auth-libs-1.4-1.Bull.noarch.rpm
```

3. Install the server package only on the nodes that supply the LDAP authentication service:

```
# rpm -ivh openldap-auth-server-1.4-1.Bull.noarch.rpm
```

4. Install the client package on all nodes that will ask for the LDAP authentication service:

```
# rpm -ivh openldap-auth-clients-1.4.1.Bull.noarch.rpm
```

These packages install tools that allow users and groups to be initialized, administrated and managed in the LDAP directory. The server package also installs specific configuration files.

The most important directories are the following:

/usr/local/ldap-auth	Contain specific scripts to facilitate the administration of the Directory for authentication purpose.
/usr/local/sbin	Contain tools to manage users, groups and SSH public keys.
/etc/openldap	Contain the configuration files of the OpenLDAP directory.
/etc/openldap/cacerts	Will contain the certificates used for TLS.
/var/lib/ldap-auth	Will contain the Database files of the Directory.

Note Since the data contained in the database are very sensitive, you must plan regular backup of the DataBase, using an appropriate backup solution.

Some tools are reserved to the administrator, while some can be used by all users registered in the directory.

Tools installed in /usr/local/ldap-auth:

auth.ldap	Activate / de-activate the authentication by LDAP system.
gen_cert.ldap	Enable the certificates required for TLS to be automatically generated.
importgroup.ldap	Import the description of groups directly from local configuration files (/etc) to the Directory.
importuser.ldap	Import the description of users directly from local configuration files (/etc) to the Directory.
init.ldap	Initialize the LDAP Directory structure.
listgroups.ldap	Display the list of the groups declared in the Directory.
listuserkeys.ldap	Display the list of the public keys of the Directory for a specific user
listusers.ldap	Display the list of the users declared in the Directory.

Tools installed in /usr/local/sbin:

addtogroup.ldap	Add a user to an existing group.
chsh.ldap	Enable the shell of a user to be modified.
delfromgroup.ldap	Remove a user from a specified group.
encryptpass.ldap	Get an encrypted password in conformance with the format required by the OpenLDAP configuration files.
groupadd.ldap	Add a group into the Directory.

groupdel.ldap	Remove a group from the Directory.
lockuser.ldap	Lock a user (the user's shell is changed to restricted shell /sbin/nologin).
passwd.ldap	Change the user password.
sshkeyadd.ldap	Add a public SSH key to the user profile.
sshkeydel.ldap	Delete a public SSH key to the user profile.
unlockuser.ldap	Unlock a user that was locked to restricted shell.
useradd.ldap	Add a user to the Directory.
userdel.ldap	Remove a user from the Directory.

Note The –h option provides help for each command. You can also refer to the man pages.

Most commands require a password. For the administrator commands this is the password defined during initialization of the Directory. For user commands enter the UNIX user password.

4.2 Configure the Master Server

The LDAP Master Server is the reference machine on which the requests are issued firstly. For data consistency reasons, the modifications of the Directory must be performed exclusively on the Master Server.

The Master Server is configured using the /etc/openldap/slapd-auth.conf configuration file.
A template file is supplied with the openldap-auth package. It contains all the directives necessary for using the LDAP server for authentication.

/etc/openldap/slapd-auth.conf Template file:

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openssh-lpk.schema

pidfile /var/run/openldap/slapd-auth.pid
argsfile /var/run/openldap/slapd-auth.args

disallow bind_anon

access to attrs=uid,cn,uidNumber,gidNumber,homeDirectory,gecos
by dn="cn=ldaproot,dc=cluster,dc=net" write
by dn="cn=ldapbind,dc=cluster,dc=net" read
by dn="cn=ldaprep,dc=cluster,dc=net" peername.ip=127.0.0.1 read
by self read
by users read
by * none

access to attrs=loginShell
by dn="cn=ldaproot,dc=cluster,dc=net" write
by dn="cn=ldapbind,dc=cluster,dc=net" read
by dn="cn=ldaprep,dc=cluster,dc=net" peername.ip=127.0.0.1 read
by self write
by users read
by * none

access to attrs=userPassword
by dn="cn=ldaproot,dc=cluster,dc=net" ssf=56 write
by dn="cn=ldaproot,dc=cluster,dc=net" peername.ip=127.0.0.1 write
by dn="cn=ldapbind,dc=cluster,dc=net" ssf=56 auth
by dn="cn=ldapbind,dc=cluster,dc=net" peername.ip=127.0.0.1 auth
by dn="cn=ldaprep,dc=cluster,dc=net" peername.ip=127.0.0.1 read
by self ssf=56 write
by self peername.ip=127.0.0.1 write
by anonymous ssf=56 auth
by anonymous peername.ip=127.0.0.1 auth
by * none

access to *
by dn="cn=ldaproot,dc=cluster,dc=net" write
by dn="cn=ldaprep,dc=cluster,dc=net" peername.ip=127.0.0.1 read
by self write
by users read
by * none

# Set limits to avoid DOS
#
limits dn="cn=ldaprep,dc=cluster,dc=net"
size.soft=unlimited size.hard=unlimited size.unchecked=unlimited
time.soft=unlimited time.hard=unlimited

limits dn="cn=ldaproot,dc=cluster,dc=net"
size.soft=unlimited size.hard=unlimited size.unchecked=unlimited
time.soft=unlimited time.hard=unlimited

limits users
size.soft=2048 size.hard=8192 size.unchecked=32767
time.soft=20 time.hard=60

limits anonymous
size.soft=2048 size.hard=4096 size.unchecked=8192
time.soft=15 time.hard=40

database bdb
cachesize 10000
suffix "dc=cluster,dc=net"
checkpoint 512 720

# logout idle clients after x seconds
idletimeout 300

# Do not uncomment rootdn and rootpw directives
#
#rootdn "cn=ldapbind,dc=cluster,dc=net"

# Generate password using slappasswd
# Default: clusterbull
#
#rootpw {SSHA}Ck/TgJ+C0riK00Y/u1RZzj/nVpSfxr1v
```

```
directory /var/lib/ldap-auth
mode 0600

security ssf=1 update_ssf=112 simple_bind=64
password-hash {SSHA}

TLSCipherSuite HIGH:MEDIUM:+TLSv1
TLSCACertificateFile /etc/openldap/cacerts/cacert.pem
TLSCertificateFile /etc/openldap/cacerts/slapd.pem
TLSCertificateKeyFile /etc/openldap/cacerts/slapd.key

index objectClass eq
index cn pres,eq
index sn pres,eq,sub
index uid pres,eq
index uidNumber,gidNumber eq
index ou,mail,givenname,memberUid eq,pres,sub
index loginShell,sshPublicKey eq,pres

# Configure replication (slurpd)
#
replica uri=ldaps://slave
binddn="cn=ldaprep,dc=cluster,dc=net"
bindmethod=simple credentials=clusterbull
replogfile /var/lib/ldap-auth/slapd-auth.replog
replica-pidfile /var/run/openldap/slurpd-auth.pid
```

Only the following parameters specific to the customer site must be modified:

- LDAP suffix

  This can be considered as the name of the Directory. By default it is set to cluster.net, and is declared as follows:

  `dc=cluster,dc=net`

  This suffix appears in several lines of the configuration file. To change it globally, use the following command (or a similar command of your choice):

```
# sed -i s/dc=cluster,dc=net/dc=mysuffix,dc=com/g /etc/openldap/slapd-auth.conf
```

  In this example the new suffix will be `dc=mysuffix,dc=com`. You can choose a name corresponding to your environment (name of your company, name of your department, etc).

- Internal administrator password

  The rootpw directive defines the internal administrator's password (not used in normal operation, only for maintenance purposes). Default value: clusterbull.

  To generate a new password:

```
# slappasswd
```

```
New password : xxxxxxxxxxx
Re-enter new password : xxxxxxxxxxx
{SSHA}5qX3+NJRJLnoLCNU9vMZ4qXai4J51qki
```

  The displayed string must be copied and pasted into the rootpw directive of the configuration file.

Important

Do not uncomment the lines that contain the rootpw and rootdn directives during normal usage of LDAP Directory. These directives should be used only for maintenance purposes.

- Replication Configuration

  If you have a Slave Server, which is highly recommended, it is necessary to configure the replication:

  – Set the URL of the Slave Server. It must correspond to the qualified name of the server (enter the **hostname** command to get this name).
    Example:
    `uri=ldaps://host.mydomain`

    WARNING
    Never use the IP address of the Server in the URL definition.

  – The password that will be used to authenticate the synchronization requests must be identical to the password that you will define at Directory initialization (**init.ldap** script) for the **ldaprep** user. It is recommended to change it now, in the **credentials** directive (by default `credentials=clusterbull`). The password is written in clear in this file.

  If you do not plan to activate the replication (no Slave Server), it is necessary to comment all the following directives:

```
# Configure replication (slurpd)
#
#replica uri=ldaps://slave
# binddn="cn=ldaprep,dc=cluster,dc=net"
# bindmethod=simple credentials=clusterbull
#replogfile /var/lib/ldap-auth/slapd-auth.replog
#replica-pidfile /var/run/openldap/slurpd-auth.pid
```

4.3 Configure the Slave Server

The Slave Server is the second LDAP server. Its role is to answer the requests when the Master Server is not able to do this. The Slave Server configuration is very similar to the Master Server configuration, except for the replication part. The replication is optional. It must be configured only if a specific server dedicated to replication is available.

1. Copy the configuration file template for the Slave Server:

```
slave # cp /etc/openldap/slapd-auth.conf.slave /etc/openldap/slapd-auth.conf
```

2. Modify the following features, as you have done for the Master Server:

   – LDAP suffix

   – Internal administrator password

   See section 4.2 *Configure the Master Server* for details.

3. Change the **master_ip** text strings:

   Access controls (ACLs) are defined to ensure the Directory security. To make them efficient on the Slave Server it is necessary to replace all the **master_ip** text strings with the real IP address of the Master Server.

   For example, if the IP address of the Master Server is `172.16.1.1`, make the changes as follows:

```
# sed -i s/master_ip/172.16.1.1/g /etc/openldap/slapd-auth.conf
```

4. Replication Configuration

   – Change the LDAP suffix for the **updatedn** directive, but do NOT change the **cn** field:

```
updatedn "cn=ldaprep,dc=cluster,dc=net"
```

   – Change the **updateref** directive to correspond to the qualified name of the Master server (enter the **hostname** command to get this name):

```
updateref ldaps://master
```

WARNING
Never use the IP address of the Server in the updateref definition.

4.4 Configure the LDAP client on the Master Server

The configuration of the LDAP client must be performed on all the cluster nodes that share information on the users. The configuration file is **/etc/ldap.conf**.

1. Declare the Master and Slave Servers:

```
host master slave
uri ldap://master:390 ldap://slave:390
```

   In these two directives replace the strings “master” and “slave” with the name of the Master and Slave servers (enter the **hostname** command to get these names).

Important

Do NOT change the port number (390), in order to avoid a conflict with another LDAP directory, such as the directory used by Lustre.

2. Change the LDAP suffix, to make it match with the suffix declared in the Server configuration part:

```
# sed -i s/dc=cluster,dc=net/dc=mysuffix,dc=com/g /etc/ldap.conf
```

3. In order to limit the access to LDAP Directory by unauthorized clients, anonymous requests are not accepted. A specific user, **ldapbind**, has been created, whose role is to send only identified requests. A password must be defined for **ldapbind** user, using the **bindpw** directive in the configuration file. It is not necessary to change it manually. This will be performed by the **init.ldap** script during Directory initialization

/etc/ldap.conf template configuration file:

```
host master slave
uri ldap://master:390 ldap://slave:390
base dc=cluster,dc=net
ldap_version 3

binddn cn=ldapbind,dc=cluster,dc=net
bindpw clusterbindpass

ssl start_tls
tls_ciphers TLSv1
tls_cacertdir /etc/openldap/cacerts
tls_cacert /etc/openldap/cacerts/cacert.pem
tls_reqcert hard
tls_checkpeer yes

timelimit 3
bind_timelimit 3
idle_timelimit 3600

pam_login_attribute uid
pam_member_attribute gid
pam_password md5

nss_base_passwd ou=People,dc=cluster,dc=net?sub
nss_base_shadow ou=People,dc=cluster,dc=net?sub
nss_base_group ou=Group,dc=cluster,dc=net?sub
```

4.5 Generate Certificates for TLS

To generate the Certificates for TLS an automatic script is available (**gen_cert.ldap**). It enables auto-signed certificates to be provided, which are generally sufficient for an LDAP configuration using TLS in a cluster environment. The procedure must be performed on both the Master and Slave servers.

Note The generation can also be done manually using openssl tools.

```
# /usr/local/ldap-auth/gen_cert.ldap
```

In the first part of the script, you are prompted to enter some information. For the **PEM pass phrase**, which will be the password used to sign the certificate, choose a complex password in order to guarantee the security of the certificates.

At the end of the script, check the data listed and answer "y" if you agree to sign and commit the certificate, as shown below.

```
Enter PEM pass phrase: xxxxxxx
Country Name (2 letter code): FR
State or province name (full name): Isere
Locality name (eg. city): Grenoble
Organization name (eg. company): Bull
Organizational unit name (eg. section): HPC
Email address: test@bull.com
. . .
Certificate Details:
Serial Number:
91:87:d1:e6:53:21:ab:e2
Validity
Not Before: Apr 27 12:57:44 2009 GMT
Not After : Jul 14 12:57:44 2017 GMT
Subject:
countryName = FR
stateOrProvinceName = Isere
organizationName = Bull
organizationalUnitName = HPC
commonName = ica0
emailAddress = test@bull.com
X509v3 extensions:
X509v3 Subject Key Identifier:
F2:F6:51:20:3E:35:1E:00:A7:17:6F:2E:9C:FF:80:46:BD:A8:9C:58
X509v3 Authority Key Identifier:
keyid:92:A5:90:CA:FE:52:51:FB:20:6A:73:FD:CC:0C:29:81:FB:FF:3B:A4
DirName:/C=FR/ST=Isere/O=Bull/OU=HPC/CN=ica0/emailAddress=test@bull.co
m
serial:91:87:D1:E6:53:21:AB:E1

X509v3 Basic Constraints:
CA:TRUE
Certificate is to be certified until Jul 14 12:57:44 2017 GMT (3000
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Note If you answer “no”, the process ends; you can restart it using the **gen_cert.ldap** script again.

All the generated files are stored in **/etc/openldap/cacerts** with the appropriate access rights.

Important

Perform the same operation on the LDAP Slave Server. The same values must be specified.

4.6 Initialize the Directory

It is necessary to initialize the content of the LDAP Directory on the Master server:

- Data structure
- Privileged users (administrator, replication, bind)
- Passwords of the privileged users.

Use the following command to initialize the LDAP Directory (the **–d** option specifies the name of the Directory; enter a name of your choice, for example the cluster name):

```
# /usr/local/ldap-auth/init.ldap -d MyCluster
```

```
Enter LDAP bind password: xxxxxxxx
Retype LDAP bind password: xxxxxxxx
```

You must enter the password defined in the **bindpw** directive of the **/etc/ldap.conf** configuration file

```
Enter LDAP administrator password: yyyyyyyyy
Retype LDAP administrator password: yyyyyyyyy
```

You must enter the password of the main Directory's administrator.

```
Enter LDAP replication password: zzzzzzzzzz
Retype LDAP replication password: zzzzzzzzzz
```

You must enter the password used for the synchronization of the Directories, as defined in the **credentials** directive of the **/etc/openldap/slapd-auth.conf** configuration file in the Master Server

```
Init LDAP directory...
```

Once the procedure is completed, three specific users are defined, each one with specific rights:

ldapbind This user is authorized to read non confidential information about users. This is required by the NSS library, which is used for authentication and limits the anonymous, non-trackable requests (bind).

ldaproot This user is the Directory Administrator (root), and is authorized to make any modification in the Directory.

ldaprep This user is used by the slurpd daemon for replication mechanism. It has read access rights on the Master Server and write access rights on the Slave Server, in order to be able to update data. It has limited ACL on the IP addresses of the servers.

4.7 Start the ldap-auth service on the Master Server

```
# service ldap-auth start
```

```
Checking configuration files for slapd: config file testing succeeded [ OK ]
Starting slapd: [ OK ]
Starting slurpd for authentication: [ OK ]
```

4.8 Check Directory Operation on the Master Server

To check that the Directory is operational on the Master Server, enter the following commands:

```
# ldapsearch -x -H ldap://localhost:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z
# ldapsearch -x -H ldap://master:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z

# ldapsearch -x -H ldap://localhost:390/ -D "cn=ldapbind,dc=mycluster,dc=com" -W -Z
# ldapsearch -x -H ldap://master:390/ -D "cn=ldapbind,dc=mycluster,dc=com" -W -Z

# ldapsearch -x -H ldap://localhost:390/ -D "cn=ldaprep,dc=mycluster,dc=com" -W -Z
# ldapsearch -x -H ldap://master:390/ -D "cn=ldaprep,dc=mycluster,dc=com" -W -Z
```

4.9 Activate the Slave Server

If you have configured a server for replication (Slave Server), it is necessary to perform the following steps so that the Master Server and the Slave Server can communicate.

4.9.1 Stop the ldap-auth service on the Master Server

To prevent any modification of the Database during the procedure, it is necessary to stop the LDAP service on the Master Server:

```
# service ldap-auth stop
```

```
Stopping slapd:
Stopping slurpd for authentication:
```

4.9.2 Copy the Directory Content

The files that contain the structure and the data of the Directory are stored in /var/lib/ldap-auth. To have the same information on the Master and Slave Servers, copy these files from the Master Server to the Slave Server:

```
master # cd /var/lib/ldap-auth
master # scp -pr * root@slave:/var/lib/ldap-auth
[...]
master # ssh root@slave chown ldap.ldap /var/lib/ldap-auth/*
master # ssh root@slave chown ldap.ldap /var/lib/ldap-auth
```

4.9.3 Concatenate the LDAP Servers Certificates

The **/etc/openldap/cacerts/cacert.pem** file must contain the certificates of the LDAP servers from which the clients will get information on the users. So it is necessary that this file contains the certificates for both Master and Slave Servers. This is done by concatenating the two certificates.

Concatenate the two certificates as follows:

```
master # scp root@slave:/etc/openldap/cacerts/cacert.pem \
/etc/openldap/cacerts/cacert.pem.slave
master # cat /etc/openldap/cacerts/cacert.pem.slave >>
/etc/openldap/cacerts/cacert.pem
master # scp /etc/openldap/cacerts/cacert.pem
root@slave:/etc/openldap/cacerts/cacert.pem
```

Note The **/etc/openldap/cacerts/cacert.pem** file will be copied on all the cluster nodes.

4.9.4 Copy the /etc/ldap.conf Configuration file on the Slave Server

Since the **/etc/ldap.conf** configuration file has been configured on the Master Server, you only have to copy it on the Slave Server:

```
master # scp /etc/ldap.conf root@slave:/etc/ldap.conf
```

4.9.5 Start the ldap-auth services in replicate mode

The service must be started on the Master Server, then on the Slave Server:

```
master # service ldap-auth start
```

```
Checking configuration files for slapd: config file testing succeeded [ OK ]
Starting slapd:                                                       [ OK ]
Starting slurpd for authentication:                                   [ OK ]
```

```
slave # service ldap-auth start
```

```
Checking configuration files for slapd: config file testing succeeded [ OK ]
Starting slapd:                                                       [ OK ]
```

The LDAP replication function is now fully operational.

4.9.6 Test the communications between Master and Slave Servers

To validate the encrypted communication between the Master and Slave Servers, and to check the consistency of data on the two Directories, enter the following commands, on the Master Server, then on the Slave Server:

```
master # ldapsearch -x -H ldap://master:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z
master # ldapsearch -x -H ldap://slave:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z
slave # ldapsearch -x -H ldap://master:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z
slave # ldapsearch -x -H ldap://slave:390/ -D "cn=ldaproot,dc=mycluster,dc=com" -W -Z
```

The output should report no error, and it should be identical on both Master and Slave Servers.

4.9.7 Activate the authentication by LDAP (PAM/NSS)

To be able to use LDAP as a source of information for users and groups, and so to submit authentication requests to LDAP, it is necessary to:

- Add the LDAP search feature in the **/etc/nsswitch.conf** file
- Change the behavior of the PAM (Pluggable *Authentication* Modules) system in the **/etc/pam.d** file
- Configure and activate the Name Service cache service via the **nscd** service.

To perform these tasks, run the **auth.ldap** script with the **–e** (enable) option, on the Master and Slave Servers:

```
# /usr/local/ldap-auth/auth.ldap -e
```

Important

Do not restart this script several times with the same option (-e or –d), otherwise the modified files (/etc/nsswitch.conf and /etc/pam.d/system-auth-ac) will not match the original configuration (without LDAP).

Note Ignore the Warning messages such as "`unable to load certificate`" or "`Expecting: TRUSTED CERTIFICATE`". These messages are not fatal; they are due to the presence of other files than the certificate in **/etc/openldap/cacerts**.

If you need to de-activate the authentication by LDAP, run the **auth.ldap** script with the **–d** option (disable)

```
# /usr/local/ldap-auth/auth.ldap -d
```

4.9.8 Test the configuration

To check that the synchronization mechanism (replication) - based on the **slurpd** daemon - works, create a **test** user on the Master Server, then check that information related to **test** user is available both on Master and Slave Servers.

1. Create the user:

```
# useradd.ldap -i
```

```
Password: xxxxxxxx

User login name: test
User home directory [/home_nfs/test]:
User real name [test]:
User ID (UID) [501]:
Group ID (GID) [502]:
User Shell [/bin/bash]:
Additional Groups (separated by ','):
OpenSSH Public Key file (hit enter to use default or generate):
Create home directory (yes/no) [yes]:
User password:
Retype user password:

Use default ~/.ssh/id_rsa.pub key for user test.

Group test (502) successfully created.

User test (501) successfully created with public SSH key managed.
```

2. Check that information related to **test** user is available both on Master and Slave Servers.

```
master # getent passwd |grep test
```

```
test:x:501:502:test:/home_nfs/test:/bin/bash
```

```
master # ssh ica0slave getent passwd |grep test
```

```
test:x:501:502:test:/home_nfs/test:/bin/bash
```

```
master # service ldap-auth stop
```

```
Stopping slapd:
Stopping slurpd for authentication:
```

```
master # getent passwd |grep test
```

```
test:x:501:502:test:/home_nfs/test:/bin/bash
```

```
master # ssh slave getent passwd |grep test
```

```
test:x:501:502:test:/home_nfs/test:/bin/bash
```

Check SSH Server and RSA public keys

To test that SSH server and RSA public keys work correctly, check that the test user can connect locally without password.

```
master # su – test
test@master $ ssh localhost
test@master $
```

Note The system needs some time to take into account the PAM behavior change and to use LDAP for authentication source. If a "`user unknown`" error occurs, try to restart the **nscd** service to accelerate the process. About 5 minutes may be needed before the **su** command is operational.

Important:
As long as the Maser Server is not accessible, the modifications in the base are not authorized and the commands that modify the Directory content generate an error. For example:

```
master # userdel.ldap -u test
```

```
LDAP error: can't connect to the directory (IO::Socket::INET: connect:
Connection refused) at /usr/local/lib64/perl5/5.8.8/x86_64-linux-
thread-multi//userLDAP.pm line 270.
```

Restart the **ldap-auth** service on the Master Server to make the commands fully operational:

```
master # service ldap-auth start
```

```
Checking configuration files for slapd: config file testing succeeded [ OK ]
Starting slapd:                         [ OK ]
Starting slurpd for authentication: [ OK ]
```

```
master # userdel.ldap -u test
```

```
The following user exists in LDAP directory:
uid : test
cn : test
uidNumber : 501
gidNumber : 502
loginShell : /bin/bash
homeDirectory : /home_nfs/test
gecos : test
others groups : test

The user will be removed from the LDAP directory, are you sure [no]: yes
```

```
master # groupdel.ldap -g test
```

```
Group test (502) has been removed from LDAP directory.
```

As the **test** user and group have been deleted, the **getent** commands no longer issue output:

```
master # getent passwd | grep test
master # getent group | grep test
master # ssh slave getent passwd |grep test
master # ssh slave getent group |grep test
```

Chapter 5. Configuring LDAP Clients

To configure the client nodes, install each SIS reference node, make an image, then deploy it. The following procedure explains how to install each reference node in order to use LDAP for authentication source.

5.1 Required Package

- Install the required packages (see 3.2 Required packages (dependencies)).
- Install the **openldap-auth-clients** specific package for LDAP authentication, as follows:

```
# rpm -ivh openldap-auth-clients-1.4-1.Bull.noarch.rpm
```

- Install the libraries package:

```
# rpm -ivh openldap-auth-libs-1.4-1.Bull.noarch.rpm
```

5.2 Copy the certificate from the Master Server

Each client needs to have access the required certificate for the connection to the secured LDAP server (TLS). So it is necessary to copy it on the client node:

```
# scp /etc/openldap/cacerts/cacert.pem root@nodeX:/etc/openldap/cacerts/cacert.pem
```

5.3 Copy the LDAP configuration file

Each client needs to have access the LDAP configuration details, which are in the **/etc/ldap.conf** file. So, it is necessary to copy it on the client node:

```
# scp /etc/ldap.conf root@nodeX:/etc/ldap.conf
```

5.4 Activate authentication by LDAP

As for the server, it is necessary to indicate to the system that LDAP must be used for authentication. This is done using the **auth.ldap** script.

```
# /usr/local/ldap-auth/auth.ldap -e
```

Note Ignore the Warning messages such as "`unable to load certificate`" or "`Expecting: TRUSTED CERTIFICATE`". These messages are not fatal; they are due to the presence of other files in **/etc/openldap/cacerts**.

Chapter 6. Tuning

When using centralized authentication mechanism such as LDAP or NIS a lot of authentication requests occur in direction of the authentication service daemon. To avoid bottlenecks that may become a big issue on large clusters a solution is to use the **Name Service Cache Daemon (nscd)**, which is included in the Linux distribution.

To activate NSCD, run these steps:

Note This procedure has to be done on all nodes (Management, I/O, Login, Compute) if they use the centralized authentication system.

1. Setup the **/etc/nscd.conf** configuration file with the following parameters:

```
        logfile                 /var/log/nscd.log
        debug-level             3
        threads                 8
        server-user             nscd
        debug-level             0
        reload-count            unlimited
        paranoia                no

        enable-cache            passwd          yes
        positive-time-to-live   passwd          600
        negative-time-to-live   passwd          20
        suggested-size          passwd          211
        check-files             passwd          yes
        persistent              passwd          yes
        shared                  passwd          yes
        max-db-size             passwd          33554432
        auto-propagate          passwd          yes

        enable-cache            group           yes
        positive-time-to-live   group           3600
        negative-time-to-live   group           60
        suggested-size          group           211
        check-files             group           yes
        persistent              group           yes
        shared                  group           yes
        max-db-size             group           33554432
        auto-propagate          group           yes

        enable-cache            hosts           no
```

Note The default configuration can work correctly, but the parameters proposed above allow a better authentication caching tuning.

2. Activate the service at boot time:

```
# chkconfig nscd on
```

3. Start the service:

```
# service nscd start
```

Chapter 7. Generating unique host keys for each node

By default all the nodes deployed from the same reference image have the same SSH node key. This configuration limits the security of the SSH protocol, since some external attacks based on the usurpation of identity of a node can occur. A supplementary security feature can be implemented to prevent this problem. It consists in generating a unique identification SSH key for each cluster node. This feature is optional but highly recommended, especially for open clusters (totally or partially reachable from an external network).

7.1 Required Package

To facilitate the generation of the SSH node keys and their deployment, a specific package is available: **gensshkeys**. Install this package on the Management Node:

```
# rpm -ivh gensshkeys-0.1-1.Bull.noarch.rpm
```

This installs three scripts in **/usr/local/sbin**:

hostkeygen Enable the generation, the deployment and the verification of the SSH node keys.

scansshkeys Provide the list of the SSH node keys for a set of nodes.

stricthostkey Activate or deactivate the **StrictHostKeyChecking** parameter in the client SSH configuration.

This package also generates a **known_hosts** global file containing all the public keys of the cluster. This allows the “**StrictHostKeyChecking yes**” option to be kept in the client SSH configuration.

hostkeygen Usage:

hostkeygen -p repository -n nodes[x-y] [-g(enerate) || -d(eploy) || -c(heck)]

-p repository	Define the directory where keys files will be stored
-n nodes[x-y]	Specify list of nodes
-g	Generate host keys
-d	Deploy generated keys on nodes
-c	Check nodes host key

7.2 Generate and deploy the keys

For example to generate the host keys for a cluster whose name is myclus , for nodes 0 to 2, enter:

```
# /usr/local/sbin/hostkeygen -p MyClus -n myclus[0-2] -g
```

```
Generate SSH host key files... /
```

To deploy the generated keys, enter:

```
# /usr/local/sbin/hostkeygen -p MyClus -n myclus[0-2] -d
```

```
OpenSSH restart on myclus1... [OK]
OpenSSH restart on myclus2... [OK]
OpenSSH restart on myclus0... [OK]
```

Note The generation and deployment can be performed simultaneously with the following command:

```
# /usr/local/sbin/hostkeygen -p MyClus -n myclus[0-2] -d -g
```

7.3 Check the installed keys

To check that the nodes use the right key, enter:

```
# /usr/local/sbin/hostkeygen -p MyClus -n myclus[0-2] -c
```

```
Current myclus0's host key is correct...
Current myclus1's host key is correct...
Current myclus2's host key is correct...
```

7.4 Deactivate the “StrictHostKeyChecking no” option on the nodes

In order to guarantee the nodes identity during SSH connections it is necessary to deactivate the “**StrictHostKeyChecking no**” option in the **/etc/ssh/ssh_config** file of all configured nodes. This is done using the **stricthostkey** command with the **–e** option (enable) and the list of the nodes to configure. For example:

```
# /usr/local/sbin/stricthostkey -n myclus[0-2] -e
```

```
StrictHostKeyChecking set on myclus0... [OK]
StrictHostKeyChecking set on myclus1... [OK]
StrictHostKeyChecking set on myclus2... [OK]
```

Chapter 8. Troubleshooting

This chapter describes some problems that might occur (mainly connection problems) and how they may be solved.

8.1 ldap_connect error

```
* ldap_connect: (TLS) ldap_start_tls(): Connect error (-11)
additional info: error: 14090086:SSL
routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
[LDAP] could not initialize ldap connection
```

This problem frequently occurs in the case of a bad synchronization of the time. Check the synchronization of the dates between the different machines, and if necessary synchronize them with the date and time of the LDAP Master Server where the certificate has been generated.

8.2 “Can't contact LDAP server” error

```
* pam_ldap: ldap_simple_bind Can't contact LDAP server
fatal: Access denied for user XXX by PAM account configuration
```

In this context there is a problem in accessing the LDAP server.

- Check that the **nscd** service is started.
- Check (in the **/etc/ldap.conf** file) the password that authorizes « bind » operations.
- Check that the read access rights are authorized for all users for the **/etc/ldap.conf** and **/etc/openldap/cacerts/cacert.pem** files.

8.3 “Invalid credentials (49)” error

```
* ldap_bind: Invalid credentials (49)
```

All the code 49 errors identify a problem of authentication on the LDAP server. Generally a wrong password has been entered either in a command line, in configuration files, or in the user settings in the Directory.

Glossary

A

ACL

Access Control List

H

HPC

High Performance Computing

L

LDAP

Lightweight Directory Access Protocol

LPK

LDAP Public Key

N

NIS

Network Information Service

NSCD

Name Service Cache Daemon

NSS

Name Service Switch

P

PAM

Pluggable Authentication Module

S

SSH

Secure Shell

T

TLS

Transport Layer Security

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A2 41FD 00