# Configuring a Virtual-Domain Server with LDAP

This document provides a recipe for configuring a Mirapoint server to perform LDAP authentication, message routing, and email access proxying. Configuration requires two activities:

- LDAP Database User Definition (see An Introduction to LDAP on page 1)
- Mirapoint Server Configuration on page 4

A Mirapoint server can be set up to perform as an IMR (inbound message router), an email access proxy, or both. It can also serve equated (aliased) domains when needed.

- Changes to Support Equated Domains on page 7

## Mirapoint LDAP Schema Requirements

Mirapoint software requires the following fields, or equivalent fields, in the LDAP database:

**User:Mailhost**

Fully-qualified name of the message server host (e.g. mail.example.com)

**User:RoutingAddr**

The user mailbox name plus the mailhost name (e.g. user@mail.example.com)

**User:PublishedName**

The canonical mail routing address (e.g. Joe.User@mail.example.com)

You must set the base distinguished name to be the same for all three, and you must also set the filter to be the same for all three.

## An Introduction to LDAP

LDAP (Lightweight Directory Access Protocol) is an Internet standard for storing information and responding to queries. RFC 1777 defines the protocol, and RFC 1558 describes search filters. Unlike some name services, LDAP is hierarchical and can accommodate anything from simple to complex multilevel databases. Although this document presents an single example scenario, LDAP is flexible, so your installation might vary radically. One way that LDAP implementations vary is in

the naming of hierarchies. Here are some examples of different LDAP database schemas:

| | |
|---|---|
| Network Domain Based | `dc=mirapoint,dc=com` |
| Geographic Based | `l=Barstow,st=California,dc=US` |
| Organization Based | `ou=Engineering,o=MyCompany` |
| Functional Categories | `ou=Marketing,o=WorldwideSales` |

LDAP entries, or objects, are uniquely named containers of user-definable attributes. While most LDAP implementations make it possible for you to strictly enforce, loosely enforce, or not even enforce which attributes can (or must) be assigned to an entry, it is considered good practice to assign an attribute-value pair to all directory objects. The example below shows one possible way to define an email user in the LDAP directory.

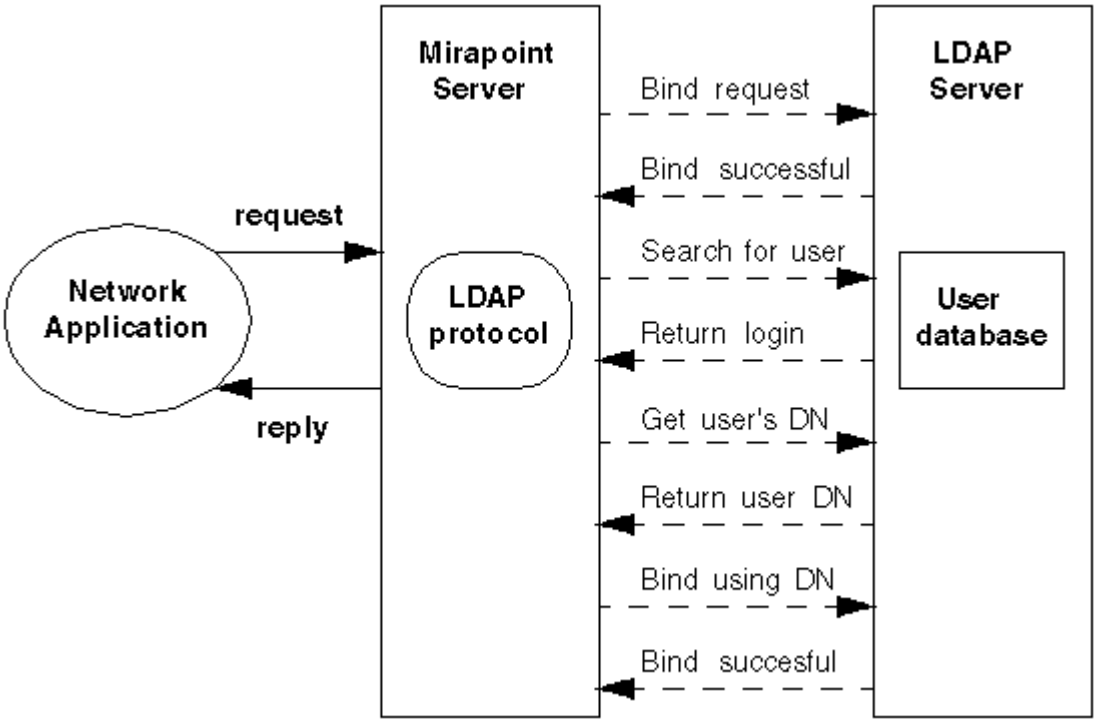


Figure 1 Getting User Information from LDAP

## Example LDAP User Definition

The LDAP service must be running on your local network. Each non-local messaging user would have a database record containing at least the following information, shown here in LDIF (LDAP data interchange format) representing the LDAP schema:

```
dn: uid=USERNAME,dc=DOMAINPART,dc=DOMAINPART,dc=DOMAINPART
uid: USERNAME
cn: COMMONNAME
mail: USERNAME@DOMAIN
userpassword: PASSWORD
mailhost: MAILSTORE-FQDN
maildrop: USERNAME@MAILSTORE-FQDN
objectclass: inetorgPerson
```

In the sample LDIF entry above, italicized words indicate variables that you provide. The last line does not change.

Mirapoint software does not require any particular LDAP schema, but uses the four fields starting with mail. The first line specifies the distinguished name (DN), an unambiguous location for the entry in the directory hierarchy. The first part of the DN is a leaf-object, unique in its branch of the directory information tree (DIT). The remaining portions of the DN are the names of non-leaf nodes leading to the directory root (similar to a filesystem).

This example uses the network-domain-based namespace. However an LDAP-aware Mirapoint server does not enforce any particular DIT structure or orientation. Many LDAP schemas follow a different syntax: instead of dc=, they use c= for country, o= for organization, and ou= for organizational unit.

- `USERNAME` is the user's login name used for authentication
- `DOMAINPART` is the domain split into period-separated components
- `COMMONNAME` is the user's first and last name, or other moniker
- `DOMAIN` is the (possibly masquerade) domain from which e-mail originates
- `PASSWORD` is the user's password (plain text or encrypted, see below)
- `MAILSTORE-FQDN` is the fully qualified domain name of the mail store server

Suppose Joe User chooses the email address juser@example.com, sets his password to verysecret, and has his inbox located on mas1.example.com. The LDIF entry for Joe should look like this:

```
dn: uid=juser,dc=example,dc=com
uid: juser
cn: Joe User
mail: juser@example.com
userpassword: verysecret
mailhost: mas1.example.com
maildrop: juser@mas1.example.com
objectclass: inetorgPerson
```

LDAP authentication is done with a Bind to the specified DN (dn: distinguished name), using the given userpassword as password. The CN (cn: common name) shows the user's full name, and mailhost indicates where the inbox is stored. The maildrop line is for future expansion, in case you want to rewrite mailboxes to a different name.

### Password Encryption

It is possible to avoid using a plain text password by specifying an encryption technique inside braces. For example, this line specifies a UNIX-style crypt(3) encrypted password. Most UNIX-based LDAP servers are capable of this.

```
userpassword: {crypt}1Abhjfjry8jf.
```

Your LDAP server must support the given encryption technique.

## Mirapoint Server Configuration

The Mirapoint server must be running system software version 1.6 or higher. After installing the Mirapoint hardware, you enter its IP address, netmask, default router, DNS servers, administration password, hostname, and domain name, as described in the *Hardware Installation and Maintenance* and *System Software Setup and Administration* manuals (see the Mirapoint Technical Library on the Mirapoint Support Website).

Once you have entered a valid TCP/IP configuration, you can telnet to the Mirapoint server and log in as Administrator. After giving the administration password, you may enter the following commands (order is unimportant). These commands continue the Joe User example above, using an LDAP server called `ldap.example.com` and a time server called `ntp.example.com`. As per the LDAP section above, the Mirapoint server is named `mas1`. The POP service enabling commands, shown in blue, are optional.

```
ldap add ldap.example.com:389
ldap add ldap2.example.com:389
smtp set masq example.com
smtp set ldaprouting on
maildom add example.com
maildom add virtualdomain.net
ntp add ntp.example.com
relay add example.com
service enable smtp
service enable imap
service enable pop
ldap setquery user:mailhost "$(dcmap)" "mail=$(login)" mailhost ""
ldap setquery user:RoutingAddr "$(dcmap)" "mail=$(login)" maildrop ""
ldap setquery user:PublishedName "$(dcmap)" "mail=$(login)" mail ""
auth set default plaintext:ldap
imap set mode ldapproxy
pop set mode ldapproxy
service start smtp
service start imap
service start pop
```

The `ldap add` command appends the given LDAP server to the Mirapoint system's list of LDAP servers. LDAP services are assigned port 389 by convention. For reliability, it is best to deploy multiple LDAP servers.

The `smtp set masq` command provides the masquerade (alias) that SMTP services insert into the return address of outgoing messages.

The `smtp set ldaprouting` command enables message routing using an external LDAP database of mail addresses and aliases.

The `maildom add` command causes SMTP services to attempt local delivery for the given mail domain. Your DNS server must contain an MX record for this mail domain. If setting up the Mirapoint server an a message router, add an MX record and `run maildom add` for each subdomain and virtual domain.

The `ntp add` command specifies a time server to help the Mirapoint system regulate its clock, using NTP (network time protocol).

The `relay add` command gives an IP network or domain name from which it is acceptable to relay email.

The `service enable` commands turn on (but do not start, until reboot) particular services.

Three `ldap setquery` commands define methods for retrieving important user information from the LDAP directory. Each command requires four arguments. The first argument contains the LDAP attribute name as it is known to the Mirapoint system. The second argument specifies the search base, or location in the directory

 

information tree (DIT) from which all searches are initiated. The third argument contains an attribute/value assertion called a search filter, which is passed to the LDAP server with the search request. The fourth argument is the the name of the attribute to be returned (by the server) with the search results. Null strings at end-of-line indicate type, currently ignored. For more information, enter help about LDAP in the Mirapoint command-line interface.

The `auth set default` command configures the Mirapoint system to perform an LDAP Simple Bind operation using a plain text (unencrypted) password "userpassword" as the bind credentials. An alternate per-user authentication method may be specified.

The `imap set mode ldapproxy` command causes the IMAP service to act as an email proxy, rather than providing access to mailboxes on the local host.

To avoid rebooting the system, initiate services with `service start` commands.

## Email Access Proxy

A Mirapoint server server can function as just an e-mail access proxy, providing distributed IMAP or POP mailboxes but not SMTP services. Here is the minimum set of commands to configure a virtual-domain proxy:

```
ldap add ldap.example.com:389
service enable imap
ldap setquery user:mailhost "$(dcmap)" "mail=$(login)" mailhost ""
ldap setquery user:RoutingAddr "$(dcmap)" "mail=$(login)" maildrop ""
ldap setquery user:PublishedName "$(dcmap)" "mail=$(login)" mail ""
auth set default plaintext:ldap
imap set mode ldapproxy
service start imap
```

## Inbound Message Router

A Mirapoint server can function as just an IMR (inbound message router), providing SMTP services but not IMAP or POP mail distribution. Here is the minimum set of commands to configure a virtual-domain IMR:

```
ldap add ldap.example.com:389
smtp set masq example.com
smtp set ldaprouting on
maildom add example.com
maildom add virtualdomain.net
relay add example.com
service enable smtp
ldap setquery user:mailhost "$(dcmap)" "mail=$(login)" mailhost ""
ldap setquery user:RoutingAddr "$(dcmap)" "mail=$(login)" maildrop ""
ldap setquery user:PublishedName "$(dcmap)" "mail=$(login)" mail ""
service start smtp
```

# Changes to Support Equated Domains

Sometimes it is necessary for a user to reside in two domains simultaneously. For example, when a company is bought, or changes its name, email should continue arriving under both names.

Suppose that `example.com` is acquired by `forinstance.com`, and Joe User must receive email for a while under either name. The LDAP schema must be updated with a new field, `mailAlternateAddr`. This change is shown in blue.

```
dn: uid=juser,dc=example,dc=com
uid: juser
cn: Joe User
mail: juser@example.com
mailAlternateAddr: juser@forinstance.com
userpassword: verysecret
mailhost: mas1.example.com
maildrop: juser@mas1.example.com
objectclass: inetorgPerson
objectclass: account
objectclass: organizationalPerson
```

Mirapoint system configuration must also be changed slightly to accommodate this change to the LDAP schema. Only the three `ldap setquery` commands change:

```
ldap setquery user:mailhost "$(dcmap)"
 "(|(mail=$(login))(mailAlternateAddr=$(login)))" mailhost ""

ldap setquery user:RoutingAddr "$(dcmap)"
 "(|(mail=$(login))(mailAlternateAddr=$(login)))" maildrop ""

ldap setquery user:PublishedName "$(dcmap)"
 "(|(mail=$(login))(mailAlternateAddr=$(login)))" mail ""
```

The (|) syntax indicates logical OR for two parenthesized expressions. (See RFC 1960 for more information about filter syntax.) Once the proper changes are made, mail sent to either address will arrive in the same mailbox. Joe User can now log in as either juser@example.com or juser@forinstance.com, using the same password, and reach the same mailbox.