AUTHORIZED DOCUMENTATION

Best Features Guide

Novell®
LDAP Proxy Server

1.0

October 2011

www.novell.com

Legal Notices

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

Contents

About This Guide

The *Novell LDAP Proxy 1.0 Best Practices Guide* discusses some good practices that can be followed while running Novell LDAP Proxy in your environment. The guide is organized into the following sections:

Audience

This guide is intended for network administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of the *Novell LDAP Proxy 1.0 Best Practices* Guide, visit the Novell Documentation Web site (http://www.novell.com/documentation/ldapproxy).

Additional Documentation

For additional Novell LDAP Proxy 1.0 documentation, refer to the following guides:

- *Novell LDAP Proxy 1.0 Administration Guide*
- *Novell LDAP Proxy 1.0 Installation Guide*
- *Novell LDAP Proxy 1.0 Linux Readme*

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

An Overview of the LDAP Proxy Server

1

Novell's LDAP Proxy 1.0 is a versatile product that is used to enhance an existing directory infrastructure. The LDAP Proxy 1.0 improves the security, the scalability, and the reliability of any directory infrastructure.

1.1 Need for the LDAP Proxy 1.0

The LDAP Proxy server helps to overcome certain handicaps that arise as a result of connecting directly to a directory server. Novell's LDAP Proxy 1.0 acts like a directory firewall by protecting the directory servers from furnishing information to clients from un-trusted networks and by regulating requests and responses based on certain parameters. The LDAP Proxy Server achieves improvements on the reliability and the scalability by balancing the load between any set of replica servers and by efficient fault tolerance mechanisms.

1.2 Features

This section gives a brief insight into the features of LDAP Proxy 1.0.

1.2.1 Ease of deployment

Novell's LDAP Proxy Server is completely transparent to a client and is extremely easy to deploy and manage.

1.2.2 LDAP Proxy 1.0 as a Load Balancer

To improve the scalability and the performance of a directory infrastructure, in stressful scenarios, Novell's LDAP Proxy 1.0 could be configured to act as a load balancer. As a load balancer, the LDAP Proxy 1.0 distributes the load among configured back-end group of servers. The distribution is governed by the algorithm configured by the user. In general, a connection based round robin algorithm based on priority is used to distribute the load. The LDAP Proxy 1.0 could also route requests based on the namespace.

1.2.3 Fault Tolerance

Novell's LDAP Proxy 1.0 groups clients based on certain criteria. Each of these groupings is identified as a role and intuitive policies, configured in the LDAP Proxy Server, govern the roles. A directory server is assumed to be unavailable because,

- The connection attempt returns an error, or
- The connection has timed out, or
- The directory server is unhealthy

In such scenarios, the LDAP Proxy 1.0

1. Detects the unavailability of the directory server
2. Identifies another server that can cater to the identified client role
3. Routes the connection to this newly identified server

1.2.4 Flexible to act as a SSL-izer and as a Forward Proxy

The LDAP Proxy 1.0 is designed to act as both,

1. A simple forward proxy that merely regulates the traffic to a directory server
2. A complex SSL-izer that imparts high security to the directory server

1.2.5 Unique message processing layer in the architecture

The LDAP Proxy 1.0's architecture comprises of a message processing layer that handles client requests based on certain effective policies. These policies define conditions, and actions based on the evaluation of the conditions. The message processor takes care of placing search restrictions, connection restrictions, and operation restrictions on incoming client requests.

Basic Functioning of LDAP Proxy

2

After the LDAP Proxy 1.0 is installed in your system, the LDAP Proxy 1.0 has to be configured before starting it. When LDAP Proxy 1.0 is installed, the proxy's rpm creates a directory called conf in your system. The configuration file is an xml file present in this conf directory. To configure the LDAP Proxy 1.0, to work in accordance with your needs, you have to edit the xml file present in the conf directory. This section of the document describes the various elements of the xml (configuration) file.

2.1 Organization of the Configuration file

Prior to analyzing how the configuration file is organized it is important to understand the basic functioning of the LDAP Proxy 1.0.

2.1.1 Basic Functioning of the LDAP Proxy

The LDAP Proxy 1.0 acts as a mediator between the network and the directory servers with the aim of improving the performance of the directory servers. It listens for requests from the network, establishes the identity of the incoming requests, based on the identity processes the client requests, decides to which back-end server group the processed request is to be forwarded and forwards the request to the identified back-end server group. The uniqueness of LDAP Proxy 1.0 lies in transforming the incoming request based on identity. To transform the incoming requests certain effective policies are applied to the requests.

2.1.2 The Configuration File

The proxy configuration should define,

- The interfaces the proxy is going to listen on; this definition should also include the protocols to be used for listening.
- The mechanism for establishing the identity of the incoming requests.
- The back-end servers to which the incoming requests could be forwarded.
- The back-end server grouping; this definition states how the proxy is to achieve load balancing.
- The policies to be applied to the incoming requests.

Proxy as a Load Balancer

3

To improve the scalability and the performance of a directory infrastructure, in stressful scenarios, Novell's LDAP Proxy 1.0 could be configured to act as a load balancer. As a load balancer, the LDAP Proxy 1.0 distributes the load among configured back-end group of servers. The distribution is governed by the algorithm configured by the user. In general, a connection based round robin algorithm based on priority is used to distribute the load. The LDAP Proxy 1.0 could also route requests based on the namespace.

3.1 Configuring the Listener

A listener is where the LDAP Proxy 1.0 listens for incoming requests. The proxy is capable of listening on multiple sources. Any number of listeners can be configured for the LDAP Proxy 1.0. The listener can be a port- secure or unsecure, or an interface.

The list-listener section of the configuration file, lists all the listeners configured for the proxy. You can add any number of listeners to this list. Each listener should have,

1. An identity to identify the listener.
2. A service section that specifies,
    - The protocol the listener is going to follow to listen for requests.
    - The port or the interface on which the listener is going to listen for requests.
3. At the least one request route policy that states where the incoming requests have to be forwarded to.

3.2 Configuring the Back-end Server

A back-end server is a directory server to which the LDAP Proxy 1.0 is connected. The LDAP Proxy 1.0 intercepts the requests to the back-end servers and transforms the requests based on certain policies and then forwards the transformed requests to the back-end servers, thereby improving the performance of the directory servers. A minimum of two back-end servers have to be configured for the LDAP Proxy 1.0. This facilitates load balancing and fault tolerance. A health check has to be periodically performed on the directory server to identify any performance degradation.

The list-backend-server section of the configuration file, lists all the back-end servers configured for the proxy. You can add any number of servers to this list. Each back-end server configuration should have,

1. An identity to identify the back-end server.

2. A service section that specifies,
    - The protocol the directory server is going to follow to receive requests.
    - The port or the interface through which the directory server would receive the requests from the LDAP Proxy 1.0.
3. A health check section that specifies the kind of request to be sent to the directory server to check if it is functioning properly. If this section is not specified then the server’s health is monitored by sending a simple bind request. If the server responds within 7200 seconds then the server is assumed to be performing normally.

3.3 Configuring the Proxy to Function as a Load Balancer

The back-end servers configured using the LDAP Proxy 1.0 could be grouped together for the purpose of balancing the load between the servers. Currently connection based load balancing is supported by the LDAP Proxy 1.0. The back-end servers that are grouped together have to be replicas of each other. In connection based load balancing the maximum number of connections to the back-end server and the capability of each server is taken into account and based on these two factors the load is shared between the back-end servers in a group.

Introduction to the Proxy Policies

4

The policies configured in the LDAP Proxy 1.0 act as the message processing centers. The policies enable the LDAP Proxy 1.0 to analyze the incoming requests and based on the rules defined while configuring the policies, the LDAP Proxy acts upon these incoming requests.

Any number of policies could be configured for the LDAP Proxy 1.0. Every policy has a rule associated with it. The rule has a condition part and an action part. For every rule, if the condition part evaluates to true then a specified action is performed else a default action is performed.

You can configure five types of policies in LDAP Proxy. They are,

1. Client Network Policy
2. Operation Restriction Policy
3. Map Schema Policy
4. Request Route Policy
5. Search Request Policy

The list-policy section of the configuration file lists all the policies configured for the LDAP Proxy 1.0. You can add any number of policies to this list. Every policy should have a rule section. This rule section comprises,

- A condition that the incoming request has to be checked for.
- An action that has to be performed on the incoming request if the condition evaluates to true.
- A default action that has to be performed if the condition evaluates to false.

Client Network Policy, Directory Firewall

5

The Client Network Policy is an optional policy. This policy is the most restrictive of all the policies and hence is applied to the incoming request before any other policy. This policy is applicable to all the clients. This policy establishes from which client the incoming request is received and based on the identity of the client decides if that request from that client has to be allowed or not. By this functionality the Client Network Policy acts as a directory firewall.

Operation Network Policy

6

The operation restriction policy is used to restrict the functions an incoming request could perform on a directory structure, based on the identity of the request. This policy is tied with the Request Route Policy that establishes the identity of an incoming request. Once the identity of the request is established, this policy checks the message type of the incoming requests. If the message type matches any of the types specified in the conditions section of the Operation Restriction Policy, then this policy rejects such requests.

Map Scheme Policy

7

The Map Schema Policy is used to enable every user to get a user specific view of the directory information. Theimportance of this policy is that the application need not change for the sake of the directory. If there is an incoming request that contains certain attributes whose naming conventions are not similar to the ones used in the directory,then this policy could be configured to change the incoming request so that the directory understands the request. The incoming request is changed by mapping the attributes and the object classes of the request to attributes and object classes that can be comprehended by the directory server.

Request Route Policy

8

This policy is used to bind the listener configuration, the back-end server configuration, and the other policies. The condition in the request route policy determines the identity of an incoming request. Based on the identity, the request route policy performs an action on the incoming request. This action could include transforming the incoming request by applying some other policy.

The request route policy has no default action. If the request route policy is unable to establish the identity of an incoming request then that request is ignored.

For every listener configured there should be at the least one request route policy associated with the listener. For each listener any number of request route policies could be configured. The most restrictive request route policy should be applied first to the request coming in through the listener.