

Novell RADIUS Services Administration Guide

www.novell.com

103-000130-001



Novell[®]

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

This product may require export authorization from the U.S. Department of Commerce prior to exporting from the U.S. or Canada.

Copyright © 1993-2002 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Nos. 5,870,739, 5,873,079; and 5884,304. U.S. and Foreign Patents Pending.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

Administration Guide

[October, 2002](#)

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

BorderManager is a trademark of Novell, Inc.

ConsoleOne is a trademark of Novell, Inc.

NDS is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare is a registered trademark of Novell, Inc., in the United States and other countries.

NLM is a trademark of Novell, Inc.

NMAS is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc., in the United States and other countries.

Novell Directory Services is a trademark of Novell, Inc.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

Preface	7
1 Novell RADIUS Services NMAS Overview and Planning	9
Overview of Novell RADIUS Services	9
RADIUS Protocol	9
RADIUS and NDS	11
Dial Access Attributes	12
Common Name Logins	12
RADIUS Accounting	12
RADIUS Audit Log	13
Token Authentication	13
Authentication Device Initialization	14
Password Generation	15
Hard Tokens	17
Soft Tokens	17
Token Authentication and NDS	18
RADIUS Proxy Services	18
RADIUS Authentication Proxy	18
RADIUS Accounting Proxy	18
2 Setting Up Novell RADIUS Services	21
Novell RADIUS Services Prerequisites	21
Installing Novell RADIUS Services	22
Upgrading From BorderManager Authentication Service 3.5 or 3.6 to Novell RADIUS Services	22
Creating a Dial Access System Object	23
Creating a Dial Access Profile Object	23
Creating a Dial Access Profile Object for PPP Service	23
Creating a Dial Access Profile Object for Telnet Service	24
Enabling a User for Dial Access Services	24
Starting Novell RADIUS Services	24
Completing Advanced Setup, Configuration, and Management Tasks	25
3 Managing Novell RADIUS Services	27
Changing RADIUS Server Options	27
Setting Up Dial Access Services	28
Creating and Setting Up Dial Access System Objects	28
Creating Dial Access Profile Objects	29
Creating and Setting Up User Objects	29
Setting Up Users and Groups for Container and Group Administration	30
Setting Up Organization and Organizational Unit Container Objects	30
Setting Up Group Objects	31
Setting Up Remote Connection Restrictions	32
Specifying Dial Access System Login Restrictions	32
Specifying Per-User Login Restrictions	33
Planning Login Policies	34
Login Policy Object	34
Login Rules	35
Setting Up Login Policies	35
Creating a Login Policy Object	35
Configuring Login Policy Rules	36
Planning RADIUS Proxy Services	36
Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to Any RADIUS Server	38
Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to a Specific RADIUS Server	38
38	
Setting Up a RADIUS Authentication Proxy as an ISP to Forward Requests to a Corporate RADIUS Server	38
Setting Up a RADIUS Authentication Proxy to Authenticate Usernames to a Search Domain	39

Managing RADIUS Proxy Services	39
Adding an NDS Context Domain Processed by Any RADIUS Server	39
Adding an NDS Context Domain Processed by a Specific RADIUS Server	40
Adding a Generic Proxy Server Domain	40
Adding a Search Domain Server Domain	40
Adding a RADIUS Accounting Proxy Domain	40
Modifying a Domain	41
Deleting a Domain	41

Preface

Novell® RADIUS Services provides users a way to remotely connect to the network using the security capabilities of the Remote Authentication Dial-In User Service (RADIUS) protocol. It also takes advantage of the easy and convenient management afforded by NDS.

This guide provides an overview of the Novell RADIUS Services technology and software. It includes instructions on how to configure and manage Novell RADIUS Services. It is written primarily for network administrators.

1

Novell RADIUS Services NMAS Overview and Planning

This chapter contains overview and planning information for Novell® RADIUS Services. This chapter contains the following sections:

- ♦ “Overview of Novell RADIUS Services” on page 9
- ♦ “RADIUS Proxy Services” on page 18

Overview of Novell RADIUS Services

Novell RADIUS Services enables remote users to dial in to NetWare® networks (version 5.1 or later) and access network information and resources such as files, databases, applications, e-mail, printing, and host/mainframe services. It maintains security by requiring users to authenticate to the NDS® database using the Remote Authentication Dial-In User Service (RADIUS) network security protocol before they can access these services.

Novell RADIUS Services is now an integrated part of Novell Modular Authentication Service (NMAS™) and the RADIUS server components are installed to the server during the NMAS installation if selected.

Dialing in as a user to a network that uses Novell RADIUS Services is not much different from dialing in to any other dial-in service, except users use the same account to access NetWare file and print services.

The Novell RADIUS Services user must know the syntax of the username:

Service_Name: NDS_UserName@ Domain

In the simplest case, *Service_Name* and *@Domain* are not required, and *NDS_UserName* is the common name of the user's object in NDS. *NDS_UserName* can be entered in distinguished name form or in common name form if you have configured Novell RADIUS Services to accept common name login. For example, if Joe is an employee in the Sales department at Acme, the common name form would be joe and the distinguished name form would be .joe.sales.acme.

If a user is using a dial-in service other than the default, the user must add the name of the service as a prefix to the username followed by a colon (for example, ppp:joe selects the Point-to-Point Protocol [PPP] service configured for user joe).

If a RADIUS proxy is used to authenticate from a remote RADIUS server, users must append an at-sign (@) followed by the proxy domain to the username (for example, joe@acme.com).

RADIUS Protocol

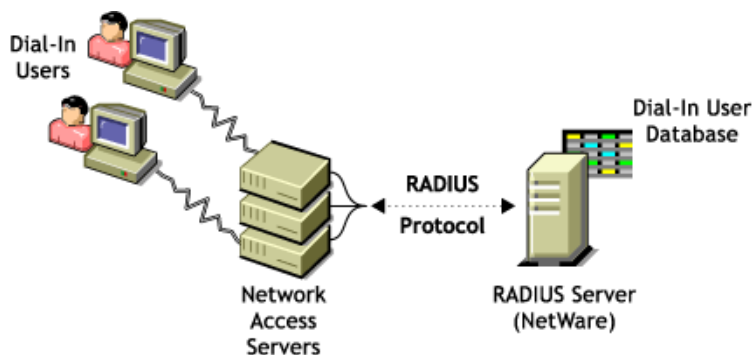
Novell RADIUS Services implements RADIUS as the network security protocol to authenticate users for dial-in remote access.

A host server running the RADIUS protocol (the RADIUS server) retrieves all dial-in user and authentication information from a central database. A host server running the RADIUS accounting protocol (the RADIUS accounting server) is responsible for logging information about dial-in user connections. The accounting information is typically used for statistical analysis, troubleshooting, and billing.

Dial-in users access the Internet or corporate intranet through network access servers, which handle communication between the users and RADIUS servers. A dial-in user must provide authentication information (typically a username and password). The network access server forwards the information to the RADIUS server using the RADIUS protocol. The RADIUS server authenticates the user by comparing the user request to the user information in the central database. The RADIUS server then returns configuration information necessary for the network access server to deliver the requested service to the dial-in user. A RADIUS server can also communicate with a RADIUS proxy server to authenticate remote users who are not in its local database.

This concept is illustrated in [Figure 1](#).

Figure 1 RADIUS Protocol



The RADIUS protocol is supported by many network access server vendors and is an Internet Engineering Task Force (IETF) Proposed Standard (RFC 2138). The RADIUS accounting protocol is also an IETF Proposed Standard (RFC 2139). The key features of the RADIUS protocol are

- ◆ Centralized administration
- ◆ Client-server model
- ◆ Network security
- ◆ Support for multiple authentication mechanisms
- ◆ User configuration and access control

Centralized Administration

The RADIUS protocol provides a central database to store all dial-in user information. This database can be used by all RADIUS-compatible network access servers.

Novell's implementation provides separate log files for system messages and accounting information. You can enable or disable the logging of system messages or accounting information from the server console. Likewise, you can specify the number of days that the log files should be maintained.

Client-Server Model

Network access servers act as clients to RADIUS servers. The network access servers authenticate users through the RADIUS server when users dial in to the network.

RADIUS servers receive user connection requests, authenticate each user, and then return all configuration information necessary for the network access server to deliver the requested service to each user.

A RADIUS server can also act as a proxy client to other RADIUS servers.

Network Security

Transactions between the network access server and the RADIUS server are protected through a shared RADIUS secret. This shared secret is never sent across the network.

In addition, user passwords are encrypted, making it difficult for outsiders to decipher them. Novell RADIUS Services utilizes login policies set up in NMAS to restrict the dial-in user account.

Support for Multiple Authentication Mechanisms

RADIUS servers support Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and other authentication mechanisms such as tokens.

Novell RADIUS Services maintains these usernames and passwords in the NDS[®] database. NMAS supports NDS-based user passwords for User objects in an NDS tree.

User Configuration and Access Control

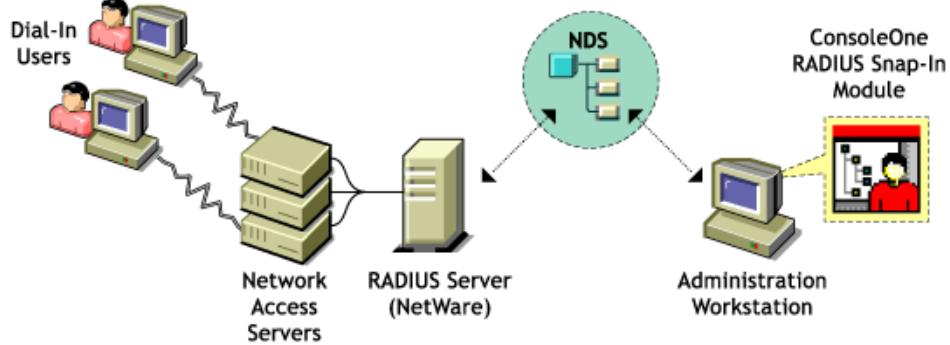
The network access server configures a user's connection and access to the network according to information provided by the RADIUS server when the user is authenticated successfully. Using Novell RADIUS Services, all users in an NDS container object can use the same configuration information or have unique dial-in configuration settings.

RADIUS and NDS

Novell RADIUS Services enables you to use the NDS application as the central database to manage all your dial-in users and services. With the RADIUS server component running on a NetWare server (version 5.1 or later), you can centrally monitor and control dial-in authentication and access to network services from the NDS database. From the administration workstation component (running on either Windows 95, 98, or Windows NT), you can centrally manage dial access services for users with the ConsoleOne™ utility. Novell RADIUS Services also enables you to take advantage of all the security, distribution, replication, and administration benefits that NDS has to offer.

This concept is shown in [Figure 2, “Novell RADIUS Services Configuration,”](#) on page 12.

Figure 2 Novell RADIUS Services Configuration



Dial Access Attributes

The RADIUS protocol defines attributes that are used to control dial-in access to the network and user configuration. When the Novell RADIUS Services server receives a request to authenticate a user, it determines whether the user is authorized to dial in (user exists, account is enabled, dial access is enabled, password is correct, and so on). If the user is authorized, the Novell RADIUS Services server constructs a list of attributes to return to the network access server to configure the user dial-in session.

The list of attributes returned depends on the following:

- ◆ The service the user specifies at login
- ◆ The attributes specified in the profile associated with the dial access service, as follows:
 - ◆ Container-specific data if the service is defined for a container object instead of a user object
 - ◆ Default dial access properties for all users in the selected container
- ◆ The type of network access server the user has dialed in to

Common Name Logins

You can configure Novell RADIUS Services to allow common name logins, as well as distinguished name logins. A common name is the name displayed in the NDS tree (such as RJONES for the user Richard Jones). A distinguished name is the complete path (or context) from the object to the root of the NDS tree (such as .RJONES.HQ.ACME.US).

You can configure common name logins by specifying a list of lookup contexts (locations of an object within the NDS tree) in the Dial Access System object. The RADIUS server searches in these locations for any user who logs in without using a distinguished name. This feature is most useful if each User object in your NDS tree has been assigned a unique common name. Users who have names that are not unique must enter their distinguished names when logging in.

Refer to the ConsoleOne online help for information about specific configuration procedures.

RADIUS Accounting

The RADIUS accounting server is responsible for logging information about dial-in user connections. The accounting information is typically used for statistical analysis, troubleshooting, and billing.

The RADIUS accounting server is typically implemented as a separate process of the RADIUS authentication server. The RADIUS accounting server listens on User Data Protocol (UDP) port number 1813. When an accounting packet is received from a RADIUS client (such as a network access server), the RADIUS accounting server logs the information in an ASCII text file and returns an acknowledgment to the RADIUS client.

When a user session begins, an accounting request packet containing connection information about a dial-in user (such as the type of service being delivered) is generated by a RADIUS client (such as a network access server) and sent to the RADIUS accounting server to be logged. When a user session ends, another accounting request packet containing the type of service delivered and any optional statistics is generated and sent to the RADIUS accounting server.

RADIUS Audit Log

The RADIUS audit log is a disk file that contains the system messages that are displayed in the RADIUS status display console. The RADIUS audit log is typically used for troubleshooting.

The RADIUS audit log file contains the same messages that are displayed on the status display (successes or failures). The status display is not required to be open in order for messages to be logged to the system log file.

The naming of the audit log file takes the form

`YYYYMMDD.LOG`

where *YYYY* represents the year, *MM* represents the month, and *DD* represents the day (for example, 19981120.LOG for an audit log file created on November 20, 1998). This convention keeps log files to a manageable size and enables you to group system log information by month, week, or day.

By default, RADIUS for NMAS starts with the RADIUS audit log file enabled. The default location for the RADIUS audit log file is as follows:

- ◆ `SYS:\ETC\RADIUS\LOG` for NetWare servers

Token Authentication

NMAS supports token-based authentication. So you can use this authentication method with Novell RADIUS Services.

Authentication is the process of determining the identity of a user attempting to access a system. The most common authentication method in use today is the static and reusable password. However, static, reusable passwords have some inherent weaknesses.

- ◆ Passwords are often simple and recognizable words and are easily guessed. This leaves them vulnerable to online guessing attacks in which an intruder keeps trying to authenticate with different passwords until discovering one that works.
- ◆ Passwords are rarely changed, resulting in default passwords that are not changed. This means that ex-employees may still have access to the system and intruders may be able to gain access at any time.
- ◆ Passwords are often shared and can be used by someone other than the assigned user.
- ◆ Passwords can be electronically stolen off the wire or stolen by rogue software on a workstation the user uses.

- ◆ Users can be tricked into divulging their passwords by trojan-horse software or a phone call by someone misrepresenting themselves.

A stronger and more effective way of identifying and authenticating remote users is through a token-based authentication mechanism. Token-based authentication implements two-factor authentication making it a much stronger authentication method. In fact, token-based authentication is often called strong authentication. With two-factor authentication, verifying a user requires two factors: something the user knows, such as a personnel identification number (PIN), and something the user has, such as the token device.

Before a user can use a token to authenticate, it must be initialized, assigned to the user, and enabled. The initialization process programs the token with the necessary profile parameters, such as the encryption algorithm and keys, the minimum and maximum PIN length, and the method of operation (synchronous or asynchronous). Once initialized, the token is assigned to a specific user and enabled.

During token authentication, the user submits a PIN to activate the token device. Invalid PINs cause the token to lock up and a special procedure is required to unlock the token. If the PIN is valid, the token uses an encryption algorithm, and a secret encryption key to encrypt a variable and generate a dynamic, one-time-use password. At the network server end, software on an authentication server uses the same encryption algorithm, secret key, and variable to generate and validate the one-time password.

The variable used to generate the one-time password is often called the challenge. There are a variety of approaches for determining the variable used to generate the password. The two main methods are usually categorized as either asynchronous or synchronous.

The asynchronous method is sometimes called the challenge-response method. With this method, the server software sends the token an external challenge—a randomly generated variable—for the token device to encrypt. The device uses this challenge variable, the encryption algorithm, and the shared secret to respond with the correctly encrypted password.

With the synchronous method, the variable is generated internally by the token. Usually, a time clock counter, a login event counter, or a combination of the two is used as the challenge variable by both the token and the server to generate the password. Because the token and the server each separately and internally determine the challenge variable, it is very important for their time clocks and the event counters to stay synchronized. If the token and server become out of sync, a special procedure is necessary to synchronize them.

Authentication Device Initialization

Before a user can use a token to authenticate, it must be initialized or programmed with the necessary profile parameters. These parameters include the algorithm and secret keys for encryption, the minimum and maximum PIN length, the language for prompts, and method of operation (synchronous or asynchronous) to be used. This initialization data must also be stored or coordinated with the authentication server. There are two methods to initialize authentication devices and store the data on the authentication server:

- ◆ Factory initialization—Authentication devices or tokens are initialized at the factory and the device images for the tokens are saved to a disk. The initialized device images are copied off the disk and imported into the authentication server's database.
- ◆ Local/manual initialization—The profile parameters are selected and entered into the local authentication server database. The server generates codes which program the token. These codes are either entered manually at the token keypad or are downloaded to a token placed in a special hardware token device initializer attached locally to the server.

When choosing between the two methods, keep the following in mind:

- ◆ The factory initialization method is the simplest, easiest, and takes the least amount of administrator effort. You import the device data, assign the devices to users and hand them out. However, with this method you are restricted to the factory configuration. This may not be ideal for your application. Also, you must trust that the factory and distributors have not compromised security either intentionally (an employee making copies of device data) or unintentionally (an intruder breaks into the factory and copies the data).
- ◆ The local/manual initialization method allows you to program a configuration that best meets your needs. Devices are initialized with random data at your site. The data is stored only in the device object and the device itself. There is no separate media (floppy disk) to protect. The data is not exposed to theft at the factory or by distributors and resellers. However, with this method each device must be programmed individually. This requires either additional hardware or manual data entry into each device.

Password Generation

Tokens use a variable as the basis to generate the one-time password. This variable is called the challenge. The two main methods for determining the variable used to generate the password are asynchronous or synchronous.

With the asynchronous or challenge-response method, the server software sends the token an external challenge—a randomly generated variable—for the token device to encrypt. The token uses this challenge variable, the encryption algorithm, and the shared secret to generate the response—the correctly encrypted password.

With the synchronous method, the challenge variable used to generate the password is determined internally by the token and the server. A time counter, event counter, or time and event counter combination within each device is used as the basis for the challenge variable. Because the token and the server each separately and internally determine the challenge variable from their own counters, it is very important for their time counters and the event counters to stay synchronized. Because it is so easy for the server and the token to get out of sync, most implementations allow for a certain amount of drift between the counters. Usually, a small range or window of these counter values is used to compute the password. However, if the token and server get out of sync beyond this window, a special procedure is necessary to synchronize them.

Both the asynchronous and synchronous methods provide a strong and effective way of authenticating users. To determine whether the synchronous or asynchronous method of operation is appropriate for your implementation, keep the following in mind:

- ◆ Synchronous method
 - ◆ The synchronous method can be used in most systems that expect a fixed password, that is, sync passwords can be entered when the GUI prompts for the password. Many of the protocols that deliver passwords to the server for validation can carry a sync one-time password and, therefore, can be used in many existing systems without changing the protocol or the user interface.
 - ◆ Synchronous passwords are easier for the user than asynchronous/challenge-response passwords because the user need not enter the challenge.
 - ◆ The synchronous method is susceptible to attacks by an intermediary who gets a valid password from the user and uses the password to falsely authenticate. The true user thinks the authentication failed and tries again successfully.
- ◆ Asynchronous/challenge-response method
 - ◆ The asynchronous method does not get out of sync.
 - ◆ The asynchronous method requires UI and protocol support that is not commonly provided in existing systems.
 - ◆ The asynchronous method is more effort for the user since the user must enter the challenge value in the token.
 - ◆ The asynchronous method makes the attack by an intermediary slightly more difficult since the authentication server generates a random challenge that cannot be predicted. Therefore, the intermediary must be more actively involved to falsely authenticate.

If you need to choose between time-based, event-based, or time- and event-based synchronous methods for your implementation, keep the following in mind:

- ◆ Synchronous method using only a time-based variable
 - ◆ As long as the clocks are accurate—don't drift from correct time—systems should remain in synchronization.
 - ◆ Passwords expire automatically and must be used immediately. They can't be written down and used later.

- ◆ Because the password is time-based, someone can capture the password and reuse it to falsely authenticate during the valid time window. This is known as the replay window.
- ◆ This method is susceptible to attacks by an intermediary.
- ◆ It may take several computations on the server side to find the right time.
- ◆ Synchronous method using only an event-based variable
 - ◆ There is no replay window. Once a password is used, it cannot be reused. The server does not allow the event counter to roll backward.
 - ◆ Someone playing with the token (generating passwords but not communicating with the server) can drive the token out of sync.
 - ◆ Passwords don't need to be used immediately. A user can generate a password, write down a list of passwords, then use the list instead of the token to authenticate. The list isn't PIN-protected and can be lost or duplicated.
- ◆ Synchronous method using a time and event-based variable
 - ◆ Passwords expire automatically
 - ◆ There is no replay window

Hard Tokens

Hard tokens are hardware-based token-generating devices. There are a variety of hardware implementations of token-generating devices. Regardless of the physical implementation of the hardware device, all hard tokens usually require users to know their PINs. Most tokens require the user to enter the PIN into the token to activate the authentication process and generate the password. Some tokens have the user enter the PIN as part of the password. The following list describes some of the more common implementations of hard tokens:

- ◆ Hand-held, calculator-sized, token device with a CPU for computing the password, memory for storing the secrets, a keypad for entering the PIN, and an LCD screen for displaying the password.
- ◆ In-line token devices that plug into the serial port on the user's PC between the PC and the modem. These devices include a CPU, and memory. The password is sent directly to the authentication server and is not displayed to the user. If there is no keypad for entering the PIN, the user must enter the PIN at the PC keyboard.
- ◆ Plug-in devices with CPU and memory that attach to the PC parallel port. The password is usually sent directly to the authentication server and is not displayed to the user. If there is no keypad for entering the PIN, the user must enter the PIN at the PC keyboard.
- ◆ Smart cards, or credit card-sized devices containing a CPU and memory. These devices are inserted into a special reader attached to the PC. They relay the passwords directly to the server. The user can enter the PIN from the PC keyboard or from the keyboard on the reader.
- ◆ Floppy disk-shaped devices that fit into the PC floppy drive. These devices contain a CPU and memory. The user enters the PIN from the PC keypad.

Soft Tokens

Soft tokens are software-based token generating devices. The software token is installed on PCs, laptops, and hand-held computers. Once the PIN is activated, the token creates and sends the user's one-time password. The system's memory stores the secrets and the system's CPU is used to generate the password. Although there is some risk associated with storing the secrets on the

system's memory, this risk is reduced by having the secrets encrypted. Also, because the token is installed on the system, anyone with physical access to the system can use it to authenticate, but they must know or guess the PIN to use it.

Token Authentication and NDS

NMAS enables you to use NDS as the central database to manage token authentication. By using ConsoleOne on an administration workstation, you can install a token login method that will allow users to do token authentication to the network either locally or remotely.

See [“Configuring Login Policy Rules” on page 36](#) for more information on how to set up token authentication.

RADIUS Proxy Services

Novell RADIUS Services can be configured to receive the following types of requests from another RADIUS server (proxy server), or to act as a RADIUS proxy and send requests to another RADIUS server:

- ◆ RADIUS authentication
- ◆ External authentication servers
- ◆ RADIUS accounting

RADIUS Authentication Proxy

A Novell RADIUS Services server can be configured to receive authentication requests from another RADIUS server (the RADIUS proxy) or to act as a RADIUS proxy and send requests to a target RADIUS server.

A user's access request (user ID and password) is sent to a proxy server if it cannot be authenticated by a local RADIUS server. The RADIUS proxy server forwards the access request to a proxy target server that can authenticate the user. The proxy target server checks the information in the user access request and either accepts or rejects the request. If the proxy target server accepts the request, it returns configuration information specifying the type of connection service (such as Point-to-Point Protocol [PPP] or Telnet) to deliver to the user.

Users who log in with a domain that has been configured to use the local NDS tree will be authenticated by the RADIUS server. Users who log in with another domain name will have their RADIUS communications sent to the proxy target server for their domain. A user can specify the target domain at login time by appending the username with an @ followed by the domain name.

NOTE: The syntax is similar to the Internet e-mail address syntax; however, the domain name is not restricted to Domain Name System (DNS), although DNS names might be convenient to use. For example, Jane, an employee of Acme, would be authenticated by the company's RADIUS server by entering her username as jane@acme.com.

RADIUS Accounting Proxy

A Novell RADIUS Services server can be configured to receive accounting requests from another RADIUS server (the RADIUS proxy) or to act as a RADIUS proxy and send accounting requests to another RADIUS server. Refer to ConsoleOne online help for information about specific configuration procedures.

2

Setting Up Novell RADIUS Services

Novell® RADIUS Services enables remote users to dial in to NetWare® networks and access network information and resources. It maintains security by requiring users to authenticate using the Remote Authentication Dial-In User Service (RADIUS) protocol. It is comprised of the following four components:

- ♦ RADIUS server (the NetWare server on which you install the RADIUS for NMAST™ software)
- ♦ Network access server (the device remote users dial in to)
- ♦ Administration workstation (ConsoleOne™)

This chapter contains the following sections:

- ♦ [“Novell RADIUS Services Prerequisites” on page 21](#)
- ♦ [“Installing Novell RADIUS Services” on page 22](#)
- ♦ [“Upgrading From BorderManager Authentication Service 3.5 or 3.6 to Novell RADIUS Services” on page 22](#)
- ♦ [“Creating a Dial Access System Object” on page 23](#)
- ♦ [“Creating a Dial Access Profile Object” on page 23](#)
- ♦ [“Enabling a User for Dial Access Services” on page 24](#)
- ♦ [“Starting Novell RADIUS Services” on page 24](#)
- ♦ [“Completing Advanced Setup, Configuration, and Management Tasks” on page 25](#)

Novell RADIUS Services Prerequisites

Before you set up Novell RADIUS Services, verify that the following prerequisites have been met:

- ♦ TCP/IP is configured and functioning on the RADIUS server and the network access server
- ♦ The network access server is RADIUS compliant (IETF RFC 2138 and RFC 2139 for accounting support)
- ♦ RADIUS authentication is enabled on the network access server
- ♦ RADIUS server address on the network access server is set to the NetWare server on which the RADIUS server software will be installed
- ♦ RADIUS secret is established and known by the network access server

Installing Novell RADIUS Services

You can choose to install the server components for Novell RADIUS Services during the NMAS installation.

If you didn't select the components during a previous NMAS installation, re-run the installation and select the RADIUS server components.

For information on installing NMAS, see the quick start card that shipped with NMAS or the INSTALL_NMAS.PDF file located on the NMAS CD.

Upgrading From BorderManager Authentication Service 3.5 or 3.6 to Novell RADIUS Services

If you are upgrading from Novell BorderManager™ Authentication Service 3.5 or 3.6 to Novell RADIUS Services, you can use the utility RADMIG.NLM to migrate your configuration information from your BorderManager Authentication Service 3.5 or 3.6 server into NDS®.

RADMIG.NLM migrates the shared secret for clients, and the shared secrets for proxy domains, and stores them in NMAS. It also takes the user's DAS password and converts it to a simple password stored in the user's object. The DAS password is converted to a simple password because a simple password is the only method that supports the CHAP protocol.

IMPORTANT: This utility will migrate the information from BMAS 3.5 or 3.6 to NDS, but once the information has been modified using ConsoleOne, the BMAS server will no longer be useful.

To migrate your configuration information from BMAS 3.5 or 3.6 to NMAS:

- 1** Load RADMIG.NLM on your RADIUS Server.
- 2** Enter the BMAS 3.5 Dial Access System object you need to upgrade by typing in the object's full distinguished name without a preceding period.
- 3** Enter the NDS password for the Dial Access System object. This unwraps the Dial Access System object's private key.
- 4** Select Yes to migrate your users' Dial Access System passwords to corresponding simple passwords in NDS.
- 5** Press Enter to search the entire NDS tree for users with a Dial Access System password, or you can search for users in a specific container by typing in the context of a specific container and pressing Enter.

The migration utility runs and migrates the passwords. You can view the changes made by the migration utility by looking at the SYS:\ETC\RADIUS\DEBUG\RADMIG.LOG file.

Creating a Dial Access System Object

An NDS Dial Access System object stores configuration information for RADIUS servers and can manage a common configuration for a collection of RADIUS servers working together. You must create at least one Dial Access System object in the NDS tree where your RADIUS server resides. All participating RADIUS servers use the Dial Access System object for configuration. The information stored in the object includes the following:

- ◆ Clients—Enables you to define IP addresses for the network access servers and the shared secrets used among the RADIUS servers, network access servers, and proxy RADIUS servers from which requests will be received.
- ◆ Domains—Enables you to configure other RADIUS servers to which you want to forward RADIUS requests.
- ◆ Username Resolution—Defines the search path for objects.
- ◆ Miscellaneous—Enables you to change the Dial Access System password and to install an attribute file.
- ◆ Remote Connection Restrictions—Enables you to restrict the number of concurrent remote connections.

To create a Dial Access System object, complete the following steps:

- 1** Start ConsoleOne.
- 2** Select an Organizational Unit container object.
- 3** Select Create and specify the name, password, and clients of the Dial Access System object.

You are now ready to create a Dial Access Profile object. Refer to the ConsoleOne online help for information about specific configuration procedures for domains and remote connection restrictions.

Creating a Dial Access Profile Object

Each Dial Access Profile object defines the common attributes of a service used by one or more dial-in users. This simplifies administration by eliminating the need to configure the attributes of each user. You can define as many profiles as required to define different services. For example, you can create a Telnet profile that enables users to connect a terminal server to a host. You can also create a Telnet profile that enables users to connect to a host using a terminal or a terminal emulation program.

The Dial Access Profile object contains a list of RADIUS attributes that specify the configuration for creating a specific service.

Creating a Dial Access Profile Object for PPP Service

To create a Dial Access Profile object for the Point-to-Point Protocol (PPP) service, complete the following steps:

- 1** Start ConsoleOne.
- 2** Select an Organizational Unit container object.

- 3 Select Create and specify the name and properties of the Dial Access Profile object for PPP service.

You can now enable users for dial access services.

Creating a Dial Access Profile Object for Telnet Service

To create a Dial Access Profile object for Telnet service, complete the following steps:

- 1 Start ConsoleOne.
- 2 Select an Organizational Unit container object.
- 3 Select Create and specify the name and properties of the Dial Access Profile object for Telnet service.

You can now enable users for dial access services.

Enabling a User for Dial Access Services

Dial access properties are added to the User object when the Novell RADIUS Services software is installed. The User Dial Access Services property page allows you to do the following:

- ◆ Enable a user for dial access services
- ◆ Select the appropriate Dial Access System for the user
- ◆ Configure or define dial-in services for the user
- ◆ Select a default service if a user is configured for more than one dial access service

In addition, the Organization and Organizational Unit Dial Access Services property pages let you define default dial access properties for all users in the selected container. You can also manage dial access services using a Group object. Refer to the ConsoleOne online help for information about specific configuration procedures.

NOTE: You can specify dial access properties that are unique to a User object on a per-property basis. This means that a User object dial access setting can override the dial access setting of the parent container object, but other settings that are not overridden in the User object will always be inherited from the parent container object.

To enable a user for dial access services, complete the following steps:

- 1 Start ConsoleOne.
- 2 Select a User object.
- 3 Select Dial Access Services and specify the dial access properties.

You can now start Novell RADIUS Services.

Starting Novell RADIUS Services

You should have performed the following tasks before you start Novell RADIUS Services:

- ◆ [Create a Dial Access System object](#)
- ◆ [Create a Dial Access Profile object](#)
- ◆ [Enable one or more User objects for dial access services](#)

To start Novell RADIUS Services on a NetWare[®] server, complete the following steps:

- 1 Enter the following command at the server console for Novell RADIUS Services:

```
LOAD RADIUS
```

TCP/IP should already be configured and running.

- 2 Enter the distinguished name of the Dial Access System object.
- 3 Enter the password of the Dial Access System object.

The following message should be displayed:

```
RADIUS services started.
```

You can now use Novell RADIUS Services.

Completing Advanced Setup, Configuration, and Management Tasks

In addition to the basic setup procedures described in this chapter, there are several advanced setup, configuration, or management procedures you might need to complete, depending on your specific configuration. See [Chapter 3, “Managing Novell RADIUS Services,” on page 27](#) for information about the following:

- ◆ Changing RADIUS server options
- ◆ Setting up dial access services and dial access attributes
- ◆ Setting up user and groups for container and group administration
- ◆ Setting up remote connection restrictions
- ◆ Planning login policies
- ◆ Setting up login policies
- ◆ Planning RADIUS proxy services
- ◆ Managing RADIUS proxy services

3

Managing Novell RADIUS Services

This chapter describes management procedures for Novell® RADIUS Services. It contains the following sections:

- ◆ “Changing RADIUS Server Options” on page 27
- ◆ “Setting Up Dial Access Services” on page 28
- ◆ “Setting Up Users and Groups for Container and Group Administration” on page 30
- ◆ “Setting Up Remote Connection Restrictions” on page 32
- ◆ “Planning Login Policies” on page 34
- ◆ “Setting Up Login Policies” on page 35
- ◆ “Planning RADIUS Proxy Services” on page 36
- ◆ “Managing RADIUS Proxy Services” on page 39

Changing RADIUS Server Options

You can change Remote Dial-In User Services (RADIUS) server options from the NetWare® server command line, including the distinguished name of the Dial Access System object and the Dial Access System password for the specified Dial Access System object.

```
LOAD RADIUS
```

```
[name = Dial Access System distinguished name] [password = Dial Access System password] [threads = number of threads] [auththreads = number of threads listening to authentication port] [port = UDP port number for RADIUS] [acctPath = RADIUS accounting directory] [fileFormat = [standard/comma] ] [rollOver = [daily/weekly/monthly] ] [serverType = [accounting/authentication] [decrementGraceLogins = [YES/NO] [affinity = NDS server]
```

All parameters are optional. The values you specify override the default values.

If you do not specify the name or password on the command line, you will be prompted to provide a name and password at startup. Names can be specified as relative distinguished names, distinguished names, or partial distinguished names. Both typed and typeless names are supported. Refer to the NDS® documentation for details on specifying names.

The default context is set to the current bindery context. After Novell RADIUS Services has been loaded, the default context is set to the Dial Access System name context.

Strings with embedded spaces must be contained in quotation marks. In addition, a quoted parameter must be preceded with a space.

The valid values for the number of threads range between 1 and 127. The default number of threads is 5, which should be satisfactory in most cases.

The default UDP port number is 1645 (the most commonly used). However, a new UDP port number (1812) has been assigned by the Internet Engineering Task Force (IETF) for RADIUS services.

The default path for the RADIUS accounting files is SYS:\ETC\RADIUS\ACCT.

The RADIUS accounting server is typically implemented as a separate process of the RADIUS authentication server. The RADIUS accounting server listens on UDP port number 1813. When an accounting packet is received from a RADIUS client (such as a network access server), the RADIUS accounting server logs the information in an ASCII text file and returns an acknowledgment to the RADIUS client.

The default RADIUS accounting file format is comma-delimited text (standard ASCII file format is optional).

The default period before a RADIUS accounting file is rolled over is daily (weekly and monthly are optional).

By default, the Novell RADIUS Services software runs both the authentication server and the accounting server when you do not specify the ServerType option on the command line. (Running just the authentication server or the accounting server is optional.)

By default, the Novell RADIUS Services software will use the local server for NDS processing. You can use the affinity command line option to specify another server. You cannot specify more than one affinity server. You cannot specify a backup NDS server.

Setting Up Dial Access Services

You must perform the following tasks to create and set up the necessary NDS objects in your NDS tree to support dial access services with Novell RADIUS Services:

- ◆ [“Creating and Setting Up Dial Access System Objects” on page 28](#)
- ◆ [“Creating Dial Access Profile Objects” on page 29](#)
- ◆ [“Creating and Setting Up User Objects” on page 29](#)

Creating and Setting Up Dial Access System Objects

You must create a Dial Access System object in your NDS tree to manage common configuration tasks for a collection of RADIUS servers working together. The information stored in this object consists of the following:

- ◆ Clients—Enables you to define IP addresses for network access servers and shared secrets among the RADIUS servers and the various network access servers.
- ◆ Domains—Enables you to configure other RADIUS servers to which you want to forward RADIUS requests.
- ◆ Username Resolution—Enables contexts to be searched when the common name portion of the username is received in an authentication request.
- ◆ Miscellaneous—Enables you to change the Dial Access System Password and install an attribute file.
- ◆ Remote connection restrictions—Enables you to limit the number of connections that a remote user can have concurrently per network.

Typically, you need only one Dial Access System object in your NDS tree.

You can easily assign rights to an NDS object using ConsoleOne™. For example, you can assign Browse and Read rights from ConsoleOne by dragging the Dial Access System object over an Organizational Unit object near the root of an NDS tree.

This section contains the following tasks:

- ◆ “Creating a Dial Access System Object” on page 29

Creating a Dial Access System Object

To create a Dial Access System object, complete the following steps:

- 1** Start ConsoleOne.
- 2** Right-click on an Organizational Unit container object.
- 3** Click New > Object > RADIUS:Dial Access System.
- 4** Specify the name, password, and clients of the Dial Access System object.

Refer to the context-sensitive help for more information.

Creating Dial Access Profile Objects

You must create at least one Dial Access Profile object in your NDS tree to define common services used by many dial-in users. The Dial Access Profile object contains a list of RADIUS dial access attributes that specify the configuration for creating a specific service.

You can set up as many profiles as you need to define different services. For example, you can create a Point-to-Point Protocol (PPP) profile that enables users to dial in and access the Internet. You can also create a Telnet profile that enables users to connect to a local host using a terminal or terminal emulator. You can specify dial access profiles in the User object that can override settings in the Dial Access Profile object.

Creating a Dial Access Profile Object

To create a Dial Access Profile object, complete the following steps:

- 1** Start ConsoleOne.
- 2** Right-click on an Organizational Unit container object.
- 3** Click New > Object > RADIUS:Profile..
- 4** Specify the name and properties of the Dial Access Profile object.

Refer to the context-sensitive help for more information.

Creating and Setting Up User Objects

The User Dial Access Services page allows you to

- ◆ Enable a user for dial access services
- ◆ Select the appropriate Dial Access System object for a user
- ◆ Configure (or define) dial-in services for a user (such as enabling a user to select one or more Dial Access Profile objects and associate user-specific settings for each)
- ◆ Select a default dial access service if a user is configured for more than one

- ◆ Configure remote connection restrictions as well as view active connections and connection history

This section contains the following tasks:

- ◆ [“Enabling a User Object for Dial Access Services” on page 30](#)
- ◆ [“Disabling Dial Access Services for a User Object” on page 30](#)

Enabling a User Object for Dial Access Services

To enable a User object for dial access services, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a User object.
- 3** Click the Dial Access Services tab and specify the dial access properties.

Refer to the context-sensitive help for more information.

Disabling Dial Access Services for a User Object

To disable a User object for dial access services, complete the following steps:

- 1** Start ConsoleOne. select the User object.
- 2** Double-click a User object.
- 3** Click the Dial Access Services tab > Disable.

Refer to the context-sensitive help for more information.

Setting Up Users and Groups for Container and Group Administration

Perform the following tasks to modify the NDS objects in your NDS tree to manage dial access services with Novell RADIUS Services:

- ◆ [“Setting Up Organization and Organizational Unit Container Objects” on page 30](#)
- ◆ [“Setting Up Group Objects” on page 31](#)

Setting Up Organization and Organizational Unit Container Objects

You can specify common dial access properties for all users in Organization or Organizational Unit container objects. The Dial Access Service page of an Organization or Organizational Unit allows you to

- ◆ Enable dial access services for all users
- ◆ Select the Dial Access System object for all users
- ◆ Configure the dial access services that can be used by all users in a container

For example, if your organization has several departments that want to allow remote users to access your corporate network, you could use BorderManager™ Authentication Services to manage users who authenticate with the RADIUS protocol. Each department could specify rights to applications, file and print services, and dial-in configuration information. However, multiple

departments could be managed by the same network administrator without the requirement to maintain multiple databases.

Specifying dial access properties in the Dial Access Service page for an Organization or Organizational Unit container object has the following benefits:

- ◆ Configuring all users in an Organization or Organizational Unit to have the same dial-in rights simplifies administration over per-user administration.
- ◆ Configuring users in different containers with different access rights enhances security.

The dial access properties that you define for an Organization or Organizational Unit container object apply to every user in the selected container object (but not to users in Organizational Units that are at a lower level in the NDS tree). Refer to the ConsoleOne online help for information about specific configuration procedures.

You can override the dial access properties of an Organization container object or Organizational Unit container object by modifying the Dial Access Services page of a User object. This allows you to specify unique dial access properties for any User object in your NDS tree.

Enabling Dial Access Services For Users in a Container Object

To enable users in an Organization or Organizational Unit container object for dial access services, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click an Organization or Organizational Unit container object.
- 3** Click the Dial Access Services tab > Enable dial access.

Refer to the context-sensitive help for more information.

Setting Up Group Objects

You can grant rights to use one or more specified Dial Access System objects to members of a Group object. Group-based administration leverages the powerful access control list (ACL) capability of NDS to enforce user dial-in access restrictions. For example, separate Dial Access System objects could be created for firewall and dial-in access servers. Then a Firewall Group object and a Dial-In Users Group object could be created with access privileges to the firewall Dial Access System object and the dial-in Dial Access System object. By making a user a member of one or both groups, access to these resources is granted selectively based on group membership. Group-based administration can also be used to allow access to high-speed connections by selected users only, while allowing low-speed connections by all users by creating multiple Dial Access System objects.

Restricting access based on assignment to a geographical region is another use for group-based administration. Dial Access System objects could be created for each geographical region that a set of users are allowed to access. Groups such as West Coast, Midwest, and East Coast could be created with users in those regions added as members. Certain users, such as sales staff, could be included in more than one geographical group to allow access to different locations.

Each Dial Access System object must have sufficient rights to any User object that can be authenticated. This can be done for multiple users in a Group object by assigning a parent container object to which the users belong to the user list of a Dial Access System object.

Likewise, the Group object must have sufficient rights to the Dial Access System object used for authentication. This can be done by assigning the Group object to the user list of the Dial Access System object.

Both of these procedures are done in the Login Policy Object's Rules tab.

Setting Up Remote Connection Restrictions

You can limit the number of connections that a remote user can have concurrently per network. You can restrict the number of concurrent dial-in connections for each User object, or you can set a default value for concurrent dial-in connections for each Dial Access System object.

This section contains the following tasks:

- ◆ [“Specifying Dial Access System Login Restrictions” on page 32](#)
- ◆ [“Specifying Per-User Login Restrictions” on page 33](#)

Specifying Dial Access System Login Restrictions

By default, the RADIUS server allows unlimited dial-in connections. You can also specify the number of concurrent dial-in connections that the RADIUS server will allow for each User object that authenticates through a given Dial Access System object.

For a given Dial Access System object, you can specify the following types of information tracked for each dial-in user:

- ◆ Timeout interval for an interim accounting packet (determines if a dial-in connection is active)
- ◆ Time interval (in days) before an entry in a user's current login connection is removed
- ◆ Maximum number of records kept in a user's login connection history

This section contains the following tasks:

- ◆ [“Setting Dial Access System Remote Connection Restrictions” on page 33](#)
- ◆ [“Setting the Current Connection Interval” on page 33](#)
- ◆ [“Setting the Interim Accounting Timeout Interval” on page 33](#)
- ◆ [“Setting the Maximum Records in the Remote Connection History” on page 33](#)

Setting Dial Access System Remote Connection Restrictions

To perform this task, you must have RADIUS accounting activated.

To add remote connection restrictions, complete the following steps:

- 1 Start ConsoleOne.
- 2 Double-click a Dial Access System object.
- 3 Click the Remote Connection Restrictions tab and specify the number of concurrent dial-in restrictions.

Refer to the context-sensitive help for more information.

Setting the Current Connection Interval

To set the current connection interval, complete the following steps:

- 1 Start ConsoleOne.
- 2 Double-click the Dial Access System object.
- 3 Click the Remote Connection Restrictions tab and specify a value (in days) for the expiring period for removing current connection entry.

Refer to the context-sensitive help for more information.

Setting the Interim Accounting Timeout Interval

To set the interim accounting timeout interval, complete the following steps:

- 1 Start ConsoleOne.
- 2 Double-click the Dial Access System object.
- 3 Click the Remote Connection Restrictions tab and enter a value (in minutes) for the interim accounting timeout.

Refer to the context-sensitive help for more information.

Setting the Maximum Records in the Remote Connection History

To add concurrent login restrictions, complete the following steps:

- 1 Start ConsoleOne.
- 2 Double-click the Dial Access System object.
- 3 Click the Remote Connection Restrictions tab.
- 4 Enter a value for the maximum records.

Refer to the context-sensitive help for more information.

Specifying Per-User Login Restrictions

You can accept the default number of concurrent dial-in connections that the RADIUS server will allow for each User object (as specified for a given Dial Access System object), or you can override the default value for a given User object to either specify a different number of concurrent dial-in connections or allow unlimited dial-in connections.

For a given User object, the following types of information are tracked automatically:

- ◆ Active connections
- ◆ Login connection history

Setting User Remote Connections Restrictions

To add remote connections restrictions, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a User object.
- 3** Select Remote Connections and specify the login restrictions.

Refer to the context-sensitive help for more information.

Planning Login Policies

All users accessing services through Novell RADIUS Services must log in and be authenticated. All authentication, regardless of which service is being accessed, is processed by NMAS™.

The login policies enforced by NMAS are defined and stored in an NDS object called the Login Policy object (LPO). The LPO contains rules that define the relationships among the services, users, and login methods so that NMAS can determine and enforce the appropriate authentication requirements.

Login Policy Object

Login rules or policies are defined and stored in NDS in the Login Policy object. Only one Login Policy object is allowed in an NDS tree and it can only be created in the Security container object. This allows you to define policies that can be used globally across multiple servers and services throughout the NDS tree.

NOTE: The policies stored in the Login Policy object apply only to RADIUS.

The Login Policy object is administered through ConsoleOne. This object enables you to set up rules that allow you to manage authentication for users logging in to Novell RADIUS Services.

You can use any login method supported by NMAS.

Login Rules

Login rules define the authentication method required for a specific user, container, or group object to access a particular NMAS service. When a user requests access, the applicable rule will be enforced. You can define a single rule for all users, or different rules for particular users. If you define multiple rules, the rules for each service type are applied in the order in which they appear in the list. Once a rule has been matched, no other rules are evaluated. To change the priority of a rule, simply change its position in the list.

You can also define the level of enforcement for a rule. The following enforcement levels are defined:

- ◆ Mandatory-The user must authenticate using this method.
- ◆ Acceptable-The user may authenticate using this method.

The following table illustrates some possible authentication rules.

Table 1 Authentication Rule Examples

Service	Users	Authentication Method	Enforcement
.DAS.Novell	.hr.acme	NDS password	Mandatory
		Simple Password	Mandatory
		SecureID	Mandatory

Setting Up Login Policies

To set up a login policy for Novell RADIUS Services, complete the following tasks:

- ◆ [“Creating a Login Policy Object” on page 35](#)
- ◆ [“Configuring Login Policy Rules” on page 36](#)

Creating a Login Policy Object

A Login Policy Object is created when you install NMAS.

If for some reason this object is deleted and you need to create a new object, complete the following steps:

- 1** Start ConsoleOne.
- 2** Select New > Object > SAS:Login Policy and click OK.

- 3 Set the properties of the Login Policy object.

Refer to the context-sensitive help for more information.

Configuring Login Policy Rules

To configure login policy rules, complete the following steps:

- 1 Start ConsoleOne.
- 2 From the Security Container, double-click the Login Policy object.
- 3 Click the Rules tab.
- 4 Click + to add a login rule.
- 5 Click Browse and select the DAS object.
- 6 On the User List tab, click + and select the user or container that you want the rule to apply to.
- 7 On the Sequences tab, click +, then select the token method, then select Mandatory.
- 8 Click OK > OK > OK.

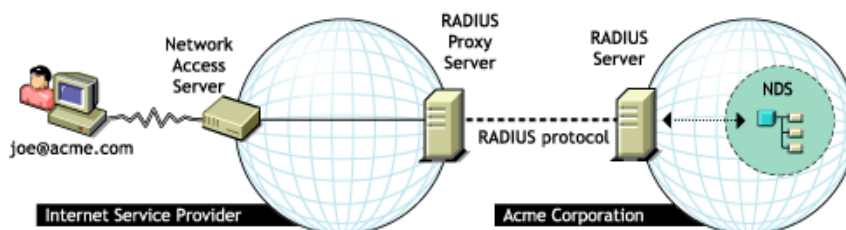
Planning RADIUS Proxy Services

You can use RADIUS proxy to out source the management of dial-in hardware to an Internet Service Provider (ISP) while you manage the users in your NDS tree. This benefit provides you with the flexibility to manage dial-in users without the investment in dial-in hardware or the burden of managing the hardware.

Using RADIUS proxy, a remote user (such as jane@acme.com) dials in to an ISP network. The user's access request (user ID and password) is forwarded to a RADIUS proxy server on the ISP network. The ISP RADIUS proxy server forwards the access request to your company's RADIUS server (such as acme.com). The RADIUS server then checks the information in the access request and either accepts or rejects the request. If the RADIUS server accepts the request, it returns configuration information specifying the type of connection service (such as PPP or Telnet) to deliver to the user.

This concept is shown in [Figure 3](#).

Figure 3 RADIUS Proxy



The RADIUS server can act as both a conventional RADIUS server and a RADIUS proxy server at the same time. To set up a RADIUS proxy, you must add a domain to the Dial Access System object's domain list. The domain name you assign is the target domain the user must use to be directed to that proxy for authentication. The RADIUS server supports usernames specified as either an NDS distinguished name or a common name. For access requests that have a username without a domain, you can configure search domains that can be checked to determine if valid

authentication information is available. The search domains consist of configured domains that do not authenticate by NDS context. Domains are defined as one of the following types:

- ◆ NDS Context—Any Novell RADIUS Services server

This domain type configures an authentication domain for the Dial Access System object that will look up users by NDS context. The authentication request can be processed by any Novell RADIUS Services server in the NDS tree. For this domain type, you specify the NDS context and define whether to look for the user in that context and any context under it, or look for the user only in the specified context. If the user is not found, you can set the option to look up the user in any defined search domains.

- ◆ NDS Context—Specific Novell RADIUS Services server

This domain type also configures an authentication domain for the Dial Access System object that will look up users by NDS context. However, this domain type will forward the authentication request to a specific Novell RADIUS Services server in the NDS tree where the user belongs to reduce network latency. For this domain type, you specify the NDS context and define whether to look for the user in that context and any context under it, or look for the user only in the specified context. The search domain option is not available. To define the target server, specify the IP address, port, and RADIUS secret of the server. To define how accounting packets are handled, specify whether to log accounting locally on the server or forward accounting packets to an accounting server on a remote domain.

- ◆ Generic proxy server

This domain type configures a simple domain proxy. Authentication requests will be forwarded to the designated RADIUS server. If the server expects to see only the common username, set the option to remove the target domain name the user logged in with. To target the server, specify the IP address, port, and RADIUS secret for the server. To define how accounting packets are handled, specify whether to log accounting locally on the server or forward accounting packets to an accounting server on a remote domain.

- ◆ Search Domain Server

This domain type configures a search domain. Search domains are searched when a user logs in with a common username (no target), or when a user with a target domain is not found in a specified NDS context and usage of a search domain is allowed for that domain. If the server expects to see only the common username, set the option to remove the target domain name the user logged in with. To target the server, specify the IP address, port, and RADIUS secret for the server. To define how accounting packets are handled, specify whether to log accounting locally on the server or forward accounting packets to an accounting server on a remote domain.

This section contains the following tasks:

- ◆ [“Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to Any RADIUS Server” on page 38](#)
- ◆ [“Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to a Specific RADIUS Server” on page 38](#)
- ◆ [“Setting Up a RADIUS Authentication Proxy as an ISP to Forward Requests to a Corporate RADIUS Server” on page 38](#)
- ◆ [“Setting Up a RADIUS Authentication Proxy to Authenticate Usernames to a Search Domain” on page 39](#)

Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to Any RADIUS Server

A user logs in as jane@acme.com. You want this user to authenticate using the local NDS tree and search for the user from the [Root] context of the NDS tree and any context below [Root]. You don't care which RADIUS server handles the authentication. If the user cannot be authenticated in the NDS tree, you want the server to send the authentication request to all the search domains for the Dial Access System object. Configure the Dial Access System object as follows:

Domain Name: acme.com

Domain Type: NDS Context—Any Novell RADIUS Services Server

NDS Context Name: [Root]

Look for user in any lookup context under this context: checked

Use search domains if user not found: checked

Refer to the context-sensitive help for information about specific configuration procedures.

Setting Up a RADIUS Authentication Proxy to Authenticate Remote Users by NDS Context to a Specific RADIUS Server

A user logs in as jane@sales.acme.com. You want this user to authenticate using the local NDS tree, but you want to search for the user only in the sales.acme context. You also want a specific RADIUS server that is within the same partition of the NDS tree as the sales context to handle the authentication to reduce network latency for the login. The IP address for the RADIUS server is 1.2.3.4 and the secret is 12345678998765432100. You need the accounting to be logged locally on the RADIUS server. Configure the Dial Access System object as follows:

Domain Name: sales.acme.com

Domain Type: NDS Context—Specific Novell RADIUS Services Server

NDS Context Name: sales.acme

Look for user in this context only: checked

Primary Address: 1.2.3.4 Port: 1645

Secret: 12345678998765432100

Log at proxy server: checked

Refer to the context-sensitive help for information about specific configuration procedures.

Setting Up a RADIUS Authentication Proxy as an ISP to Forward Requests to a Corporate RADIUS Server

You manage an ISP. Acme Corporation user joe dials in with the username joe@acme.com, and you need to forward the authentication request to the corporation's RADIUS server at IP address 1.2.3.4, port 1645, with a RADIUS secret of 12345678998765432100. You also need to forward accounting to the Acme corporation RADIUS accounting server at IP address 1.2.4.5, port 1646, with a RADIUS secret of 98765432112345678900 and a retry limit of 24 hours. Configure the Dial Access System object as follows:

Domain Name: acme.com

Domain Type: Generic Proxy

Primary Address: 1.2.4.5 Port: 1645

Secret: 12345678998765432100

Forward to domain: checked

Use alternate addresses/secret: checked
Primary Address: 1.2.4.5 Port: 1646
Secret: 98765432112345678900

Refer to the context-sensitive help for information about specific configuration procedures.

Setting Up a RADIUS Authentication Proxy to Authenticate Usernames to a Search Domain

Acme Corporation has a legacy RADIUS server. You want to migrate your remote access to NMAS and Novell RADIUS Services; however, you want to do it gradually, moving one department a month from the legacy system to NMAS and Novell RADIUS Services. You want your users to authenticate to the RADIUS server, and you want this server to search the legacy RADIUS server if the user does not exist in NDS.

To allow users to authenticate, you can set up a search domain on the NMAS RADIUS server. The legacy RADIUS server, RAD1, is at IP address 1.2.3.4, port 1645, with a secret of 09876543211234567890. You also want accounting to be logged at the legacy proxy server. Configure the Dial Access System object on the NMAS RADIUS server as follows:

Domain Name: RAD1
Domain Type: Search Domain Server
Primary Address: 1.2.3.4
Port: 1645
Secret: 09876543211234567890
Accounting Log at proxy server: checked

Refer to the ConsoleOne online help for information about specific configuration procedures.

Managing RADIUS Proxy Services

Using the ConsoleOne utility on the administration workstation, you can perform the following management tasks for RADIUS proxy services:

- ◆ [“Adding an NDS Context Domain Processed by Any RADIUS Server” on page 39](#)
- ◆ [“Adding an NDS Context Domain Processed by a Specific RADIUS Server” on page 40](#)
- ◆ [“Adding a Generic Proxy Server Domain” on page 40](#)
- ◆ [“Adding a Search Domain Server Domain” on page 40](#)
- ◆ [“Adding a RADIUS Accounting Proxy Domain” on page 40](#)
- ◆ [“Modifying a Domain” on page 41](#)
- ◆ [“Deleting a Domain” on page 41](#)

Adding an NDS Context Domain Processed by Any RADIUS Server

To add an NDS context domain to be processed by any RADIUS server, complete the following steps:

- 1 Start ConsoleOne.
- 2 Double-click a Dial Access System object.

- 3** Click Domains > Add.
- 4** Specify the domain name and select the domain type NDS Context - Any RADIUS Server.
- 5** Fill in the remaining information.

Refer to the context-sensitive help for more information.

Adding an NDS Context Domain Processed by a Specific RADIUS Server

To add an NDS context domain to be processed by a specific RADIUS server, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click the Dial Access System object.
- 3** Click Domains > Add.
- 4** Specify the domain name and select the domain type NDS Context - Specific RADIUS Server.
- 5** Fill in the remaining information.

Refer to the context-sensitive help for more information.

Adding a Generic Proxy Server Domain

To add a generic proxy server domain, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a Dial Access System object.
- 3** Click Domains > Add.
- 4** Specify the domain name and select the domain type Generic Proxy Server.
- 5** Fill in the remaining information.

Refer to the context-sensitive help for more information.

Adding a Search Domain Server Domain

To add a search domain server domain, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a Dial Access System object.
- 3** Click Domains > Add.
- 4** Specify the domain name and select the domain type Search Domain Server.
- 5** Fill in the remaining information.

Refer to the context-sensitive help for more information.

Adding a RADIUS Accounting Proxy Domain

To add a RADIUS accounting proxy domain, complete the following steps:

- 1** Start ConsoleOne.

- 2** Double-click a Dial Access System object.
- 3** Click Domains > Add.
- 4** Specify the domain name and select the domain type Generic Proxy Server or Search Domain Server.
- 5** Click the Accounting tab and fill in the appropriate information.

Refer to the context-sensitive help for more information.

Modifying a Domain

To modify a domain, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a Dial Access System object.
- 3** Click Domains and click on a domain.
- 4** Click Modify and modify the properties of the domain.

Refer to the context-sensitive help for more information.

Deleting a Domain

To delete a domain, complete the following steps:

- 1** Start ConsoleOne.
- 2** Double-click a Dial Access System object.
- 3** Click Domains and click on a domain.
- 4** Click Delete.

Refer to the context-sensitive help for more information.

