Copyright and Trademarks

© 1999 Lucent Technologies. All rights reserved.

PortMaster, ComOS, and ChoiceNet are registered trademarks of Lucent Technologies, Inc. RADIUS ABM, PMVision, IRX, and PortAuthority are trademarks of Lucent Technologies, Inc. All other marks are the property of their respective owners.

Disclaimer

Lucent Technologies, Inc. makes no express or implied representations or warranties with respect to the contents or use of this manual, and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Lucent Technologies, Inc. further reserves the right to revise this manual and to make changes to its content at any time, without obligation to notify any person or entity of such revisions or changes.
## Contents

RADIUS Installation and Configuration ........................................... 1-10

2. Configuring a RADIUS Server

   Getting Started ........................................................................... 2-1
   Selecting a RADIUS Server Host .............................................. 2-1
   Determining a Shared Secret .................................................... 2-2
   Installing RADIUS on a UNIX Host ........................................ 2-3
   Installation with pminstall ....................................................... 2-3
   Installation without pminstall ................................................. 2-4

3. Adding a RADIUS Client

   Modifying the clients File ....................................................... 3-1
   Configuring the PortMaster Using the Command Line Interface ...... 3-2
   Configuring the PortMaster Using PMVision ............................ 3-4

4. Configuring User Information

   User Profile Format ................................................................. 4-2
   Matching User Profiles ........................................................... 4-3
   Editing User Profiles ............................................................... 4-4
   Default User Profiles ............................................................... 4-4
   Check Items ............................................................................. 4-6
   Passwords ................................................................................. 4-9
   Password Encryption ............................................................... 4-11
   Username Prefixes and Suffixes .............................................. 4-12
   Called-Station-Id ..................................................................... 4-14
   Calling-Station-Id ..................................................................... 4-14
   Client Information .................................................................. 4-15
   Connect-Rate ........................................................................... 4-15
   Framed-Protocol ..................................................................... 4-16
   Group ...................................................................................... 4-16
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service-Type</td>
<td>4-17</td>
</tr>
<tr>
<td>Reply Items</td>
<td>4-19</td>
</tr>
<tr>
<td>Specifying the Type of Service</td>
<td>4-22</td>
</tr>
<tr>
<td>Enabling and Disabling Compression</td>
<td>4-26</td>
</tr>
<tr>
<td>Specifying an IP Address for the User</td>
<td>4-27</td>
</tr>
<tr>
<td>Applying a Subnet Mask to the Address</td>
<td>4-27</td>
</tr>
<tr>
<td>Adding a Route to the PortMaster Routing Table</td>
<td>4-28</td>
</tr>
<tr>
<td>Configuring RIP on the User’s Interface</td>
<td>4-29</td>
</tr>
<tr>
<td>Configuring an IPX Network Connection</td>
<td>4-29</td>
</tr>
<tr>
<td>Applying Packet Filters</td>
<td>4-30</td>
</tr>
<tr>
<td>Configuring a Login User</td>
<td>4-31</td>
</tr>
<tr>
<td>Granting a User Outbound Telnet Access</td>
<td>4-34</td>
</tr>
<tr>
<td>Setting Timeouts</td>
<td>4-35</td>
</tr>
<tr>
<td>Using Menus</td>
<td>4-37</td>
</tr>
<tr>
<td>Controlling the Number of Available Ports</td>
<td>4-38</td>
</tr>
<tr>
<td>Using RADIUS with PAP and CHAP</td>
<td>4-38</td>
</tr>
<tr>
<td>PAP</td>
<td>4-38</td>
</tr>
<tr>
<td>CHAP</td>
<td>4-39</td>
</tr>
<tr>
<td>Configuring Database Caching of User Profiles</td>
<td>4-40</td>
</tr>
<tr>
<td>Example PPP User Profile</td>
<td>4-40</td>
</tr>
</tbody>
</table>

## 5. Configuring RADIUS Menus

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu File Format</td>
<td>5-1</td>
</tr>
<tr>
<td>Single-Level Menu</td>
<td>5-2</td>
</tr>
<tr>
<td>Nested Menus</td>
<td>5-3</td>
</tr>
<tr>
<td>Termination Menus</td>
<td>5-4</td>
</tr>
<tr>
<td>Menus Called by Reference</td>
<td>5-4</td>
</tr>
<tr>
<td>Menu Filenames</td>
<td>5-4</td>
</tr>
</tbody>
</table>
### 6. Installing and Configuring ActivCard

Overview of ActivCard Components ........................................... 6-2  
How ActivCard Works with RADIUS ....................................... 6-3  
Installing the ActivEngine Components on a UNIX Host .......... 6-4  
Using the ActivEngine Test Utility ...................................... 6-6  
RADIUS Configuration for ActivCard ................................. 6-7  
  Example config.aeg File .............................................. 6-7  
Troubleshooting ActivCard .............................................. 6-9

### 7. Installing and Configuring SecurID

Overview of SecurID Components ........................................... 7-2  
How SecurID Works with RADIUS ....................................... 7-3  
ACE/Server Installation on a UNIX Host ............................ 7-4  
  Getting Started ...................................................... 7-4  
  Installing ACE/Server and Client Software on a UNIX Host .. 7-6  
  Administering ACE/Server with sdadmin .......................... 7-7  
  Authenticating with sdshell ......................................... 7-8  
RADIUS Configuration for SecurID ...................................... 7-10  
  PIN Assignment ..................................................... 7-10  
  Entering an Invalid Token Code ................................. 7-12  
Troubleshooting SecurID .............................................. 7-13

### 8. Implementing RADIUS Accounting

How RADIUS Accounting Works ............................................ 8-1  
Getting Started .......................................................... 8-3  
Client Configuration ...................................................... 8-4  
Accounting Server Configuration ......................................... 8-4  
  Installation .......................................................... 8-4  
  Configuring Options .................................................. 8-5
# Contents

Accounting Attributes ...................................................... 8-6  
  Acct-Authentic ............................................................. 8-6  
  Acct-Delay-Time ........................................................... 8-6  
  Acct-Input-Octets and Acct-Output-Octets .......................... 8-6  
  Acct-Session-Id ............................................................ 8-6  
  Acct-Session-Time .......................................................... 8-7  
  Acct-Status-Type ........................................................... 8-7  
  Acct-Terminate-Cause ...................................................... 8-7  
  Timestamp ................................................................. 8-9  
  Called-Station-Id and Calling-Station-Id ............................ 8-9  
  LE-Advice of Charge and LE-Terminate-Detail ....................... 8-9  
  NAS-Port-Type ............................................................. 8-10  
  Request-Authenticator ..................................................... 8-10  
  Start and Stop Records ................................................... 8-11  
  Example 1 ........................................................................ 8-11  
  Example 2 ........................................................................ 8-13  

9. Configuring RADIUS Proxy Service  
   How Proxy Service Works .................................................... 9-2  
   Servers Running Proxy Service .......................................... 9-5  
   Proxy Confederations ....................................................... 9-6  
   Configuring Proxy Information on the Server ....................... 9-7  
      Components of the proxy File ......................................... 9-7  
      Special Realms: DEFAULT and NOREALM ......................... 9-9  
   On the Forwarding Server ............................................... 9-10  
   On the Remote Server ..................................................... 9-10  
   Example Proxy Server Relationships and Configuration Steps .... 9-11
A. Troubleshooting RADIUS

Troubleshooting RADIUS Authentication .............................................. A-1
  Checking the radiusd Daemon ......................................................... A-1
  Checking the PortMaster .............................................................. A-1
  Checking /etc/raddb/users ............................................................. A-2
  Host Unavailable ........................................................................... A-3
  Invalid Login after 30-Second Wait ................................................. A-4
  Turning on the Debug Function ......................................................... A-5
  Result of Debug Output ................................................................ A-6
  Performance Degradation ................................................................. A-8

Troubleshooting RADIUS Accounting .................................................. A-8

B. RADIUS for UNIX Error Messages

  Accounting ............................................................... B-1
  Authentication ............................................................... B-3
  Clients .............................................................................. B-9
  Dictionary ........................................................................... B-11
  Menu .................................................................................. B-14
  Miscellaneous ................................................................. B-14
  SecurID ............................................................................... B-16
  Users ................................................................................. B-17

C. RADIUS Options

D. RADIUS Dictionary

E. Contact Information for Third-Party Products

  ActivCard ............................................................... E-1
  Website ................................................................. E-1
  Voice ................................................................. E-1
  Fax ............................................................................... E-1
Email ................................................................. E-2
iPass ............................................................... E-2
Website ............................................................. E-2
Voice ............................................................... E-2
Fax ................................................................. E-2
Email ............................................................... E-2
SecurID ............................................................. E-3
Website ............................................................. E-3
Voice ............................................................... E-3
Fax ................................................................. E-4
Email ............................................................... E-4

Index
About This Guide

The *RADIUS for UNIX Administrator's Guide* provides complete installation, configuration, and troubleshooting instructions for the Remote Authentication Dial-In User Service (RADIUS) invented by the Remote Access Business Unit of Lucent Technologies, Inc.—formerly Livingston Enterprises, Inc. This guide covers RADIUS server release 2.1 for UNIX platforms.

RADIUS can be used with the PortMaster® family of products available from Lucent Remote Access, as well as with the ChoiceNet® client/server packet-filtering software.

To install and configure these products, see “PortMaster Documentation.”

Audience

This guide is designed to be used by qualified system administrators and network managers. Knowledge of UNIX and basic networking concepts is required to successfully install RADIUS for UNIX. If you use RADIUS with third-party products—such as ActivCard or SecurID—you also must be familiar with their installation, configuration, and use.

PortMaster Documentation

The following manuals are available from Lucent Remote Access. The hardware installation guides are included with most PortMaster products; other manuals can be ordered through your PortMaster distributor or directly from Lucent.

The manuals are also provided as PDF and PostScript files on the PortMaster Software CD shipped with your PortMaster.

In addition, you can download PortMaster information and documentation from [http://www.livingston.com](http://www.livingston.com).

- *ChoiceNet Administrator’s Guide*
  
  This guide provides complete installation and configuration instructions for ChoiceNet server software.
• **PortMaster Command Line Reference**
  This reference provides the complete description and syntax of each command in the ComOS command set.

• **PortMaster Configuration Guide**
  This guide provides a comprehensive overview of networking and configuration for PortMaster products.

• PortMaster hardware installation guides
  These guides contain complete hardware installation instructions. An installation guide is shipped with each PortMaster.

• **PortMaster Routing Guide**
  This guide describes routing protocols supported by PortMaster products, and how to use them for a wide range of routing applications.

• **PortMaster Troubleshooting Guide**
  This guide can be used to identify and solve software and hardware problems in the PortMaster family of products.

• **RADIUS for UNIX Administrator’s Guide**
  This guide provides complete installation and configuration instructions for Lucent Remote Authentication Dial-In User Service (RADIUS) software on UNIX platforms.

• **RADIUS for Windows NT Administrator’s Guide**
  This guide provides complete installation and configuration instructions for Lucent Remote Authentication Dial-In User Service (RADIUS) software on the Microsoft Windows NT platform.

**ActivCard Documentation**

If you are using the ActivEngine and ActivAdmin products from ActivCard, Inc., refer to the current manual set for your software version.

For UNIX, the current manuals are

• **ActivEngine for UNIX Systems Installation and Administration Guide**

• **ActivAdmin for UNIX Systems User Guide**
Contact ActivCard for the manuals. Additionally, ActivCard might have application notes that you might find useful. See Appendix E, “Contact Information for Third-Party Products,” for information on how to contact ActivCard, Inc.

**Security Dynamics Documentation**

If you are using the ACE/Server security software from Security Dynamics, Inc., refer to the current manual set for your software version.

For UNIX, the current manuals are

- *ACE/Server v 2.3 Installation Guide*
- *ACE/Server v 2.3 for UNIX Administration Manual*

Contact Security Dynamics for the manuals. See Appendix E, “Contact Information for Third-Party Products,” for information on how to contact Security Dynamics, Inc.

**Additional References**

**RFCs**

To find a Request for Comments (RFC) online, visit the website of the Internet Engineering Task Force (IETF) at [http://www.ietf.org/](http://www.ietf.org/).

RFC 768, *User Datagram Protocol*
RFC 791, *Internet Protocol*
RFC 1321, *The MD5 Message-Digest Algorithm*
RFC 1700, *Assigned Numbers*
RFC 1824, *Requirements for IP Version 4 Routers*
RFC 1825, *Security Architecture for the Internet Protocol*
RFC 1826, *IP Authentication Header*
RFC 1827, *IP Encapsulating Payload*
RFC 1828, *IP Authentication Using Keyed MD5*
RFC 1829, *The ESP DES-CBC Transform*
RFC 2138, *Remote Authentication Dial In User Service (RADIUS)*
RFC 2139, *RADIUS Accounting*
Document Conventions

Books


Document Conventions

The following conventions are used in this guide:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
<th>Examples</th>
</tr>
</thead>
</table>
| Bold font      | Indicates a user entry—a command, menu option, button, or key—or the name of a file, directory, or utility, except in code samples. | • Enter version to display the version number.  
• Press Enter.  
• Open the permit_list file. |
### Document Advisories

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
<th>Examples</th>
</tr>
</thead>
</table>
| *Italic font*  | Identifies a command-line placeholder. Replace with a real name or value. | - set *Ether0* address *ipaddress*
                  |                                                                      | - Replace *Area* with the name of the OSPF area.              |
| Square brackets ([ ])) | Enclose optional keywords and values in command syntax. | - set nameserver [2] *ipaddress*
                  |                                                                      | - set *S0* destination *ipaddress* [ipmask]         |
| Curly braces ({ }) | Enclose a required choice between keywords and/or values in command syntax. | set syslog Logtype [{disabled] [Facility.Priority}] |
| Vertical bar (| Separates two or more possible options in command syntax. | - set *S0*|W1 ospf on|off
                  |                                                                      | - set *S0* host
                  |                                                                      |   default|prompt|ipaddress |

**Document Advisories**

**Note** – means take note. Notes contain information of importance or special interest.

**Caution** – means be careful. You might do something—or fail to do something—that results in equipment failure or loss of data.

**Warning** – means danger. You might do something—or fail to do something—that results in personal injury or equipment damage.
Contacting Lucent Remote Access Technical Support

The PortMaster comes with a 1-year hardware warranty.

For all technical support requests, record your PortMaster ComOS version number and report it to the technical support staff or your authorized sales channel partner.

New releases and upgrades of PortMaster software are available by anonymous FTP from ftp://ftp.livingston.com/pub/le/.

In North America you can schedule a 1-hour software installation appointment by calling the technical support telephone number listed below. Appointments must be scheduled at least one business day in advance.

For the EMEA Region

If you are an Internet service provider (ISP) or other end user in Europe, the Middle East, Africa, India, or Pakistan, contact your local Lucent Remote Access sales channel partner. For a list of authorized sales channel partners, see the World Wide Web at http://www.livingston.com/International/EMEA/distributors.html.

If you are an authorized Lucent Remote Access sales channel partner in this region, contact the Lucent Remote Access EMEA Support Center Monday through Friday between the hours of 8 a.m. and 8 p.m. (GMT+1), excluding French public holidays.

- By fax, dial +33-4-92-92-48-40.
- By electronic mail (email) send mail to emea-support@livingston.com.

For North America, Latin America, and the Asia Pacific Region

Contact Lucent Remote Access Monday through Friday between the hours of 7 a.m. and 5 p.m. (GMT –8).

- By voice, dial 800-458-9966 within the United States (including Alaska and Hawaii), Canada, and the Caribbean, or +1-925-737-2100 from elsewhere.
- By fax, dial +1-925-737-2110.
- By email, send mail as follows:
PortMaster Training Courses

Lucent Remote Access offers hands-on, technical training courses on PortMaster products and their applications. For course information, schedules, and pricing, visit the Lucent Remote Access website at http://www.livingston.com/tech/training/index.html.

Subscribing to PortMaster Mailing Lists

Lucent Remote Access maintains the following Internet mailing lists for PortMaster users:

- **portmaster-users**—a discussion of general and specific PortMaster issues, including configuration and troubleshooting suggestions. To subscribe, send email to majordomo@livingston.com with subscribe portmaster-users in the body of the message.

  The mailing list is also available in a daily digest format. To receive the digest, send email to majordomo@livingston.com with subscribe portmaster-users-digest in the body of the message.

- **portmaster-radius**—a discussion of general and specific RADIUS issues, including configuration and troubleshooting suggestions. To subscribe, send email to majordomo@livingston.com with subscribe portmaster-radius in the body of the message.

  The mailing list is also available in a daily digest format. To receive the digest, send email to majordomo@livingston.com with subscribe portmaster-radius-digest in the body of the message.

- **portmaster-announce**—announcements of new PortMaster products and software releases. To subscribe, send email to majordomo@livingston.com with subscribe portmaster-announce in the body of the message. All announcements to this list also go to the portmaster-users list. You do not need to subscribe to both lists.
Introduction to RADIUS

The Remote Authentication Dial-In User Service (RADIUS) is a client/server security protocol created by Lucent Remote Access. RADIUS is an Internet draft standard protocol. See RFCs 2138 and 2139 for more information on RADIUS.

User profiles are stored in a central location, known as the RADIUS server. RADIUS clients (such as a PortMaster communications server) communicate with the RADIUS server to authenticate users. The server specifies back to the client what the authenticated user is authorized to do. Although the term RADIUS refers to the network protocol that the client and server use to communicate, it is often used to refer to the entire client/server system.

Overview of RADIUS Features

RADIUS offers the following features:

- **Security**
  
  In large networks, security information can be scattered throughout the network on different devices. RADIUS allows user information to be stored on one host, minimizing the risk of security loopholes. All authentication and access to network services is managed by the host functioning as the RADIUS server.

- **Flexibility**
  
  RADIUS software from Lucent Remote Access—version 2.0 and higher—can be used with any communications server that supports the RADIUS protocol as long as you own at least one Lucent Remote Access PortMaster. For more information see the Software License Agreement at [http://www.livingston.com/license.html](http://www.livingston.com/license.html).

  RADIUS server software for UNIX platforms is distributed in source code format to Lucent Remote Access PortMaster customers. Using modifiable “stubs,” RADIUS can be adapted to work with existing security systems and protocols. You adapt the RADIUS server to your network, rather than adjusting your network to work with RADIUS.
**How RADIUS Works**

**Note** – Lucent Remote Access does not support modified RADIUS code.

- **Simplified management**
  The RADIUS server stores security information in text files at a central location; you add new users to the database or modify existing user information by editing these text files.

- **Extensive auditing capabilities**
  RADIUS provides extensive accounting trail capabilities, referred to as **RADIUS accounting**. Information collected in a log file can be analyzed for security purposes or used for billing.

The RADIUS for UNIX server version 2.1 is available in ready-to-run binary form for the following operating systems:

- Alpha Digital UNIX 4.0
- BSD/OS 2.0 and 3.0
- HP-UX 10.20
- IBM RS6000 AIX 4.2
- Redhat Linux 5.0
- SGI IRIX 6.3
- Slackware Linux 2.0.30
- Solaris 2.5.1
- Solaris x86 2.5.1
- SunOS 4.1.4

**How RADIUS Works**

RADIUS performs three primary functions. The RADIUS for UNIX server version 2.1 adds enhancements for ease of use.
Basic RADIUS Functions

The primary functions of RADIUS are authentication, authorization, and accounting. Figure 1-1 shows the authentication and authorization process.

- Authentication

  RADIUS determines whether users are eligible to receive requested services. Authentication information is stored in either a local users file or database cache, or accessed from external authentication mechanisms such as a UNIX password file, ActivCard ActivEngine database, or SecurID ACE/Server database.

Figure 1-1  RADIUS Authentication and Authorization

For example, when user bob attempts to log in to a PortMaster, the following authentication sequence takes place:

1. The PortMaster prompts bob for his username and password, and then compares the username-password pair to the PortMaster user table.

2. The PortMaster sends an access-request message to the RADIUS server if the following conditions are met:
   - Username is not found in the user table
   - Security for the port is set to on
   - RADIUS settings are configured on the PortMaster
The access-request message contains the information necessary for the RADIUS server to authenticate the user.

3. The RADIUS server checks its users file to determine if an entry for user bob is present. For bob's login to be successful, a matching username or DEFAULT entry must be found.

4. User bob is either accepted or rejected:
   - If a matching entry is found in the RADIUS users file, if the password requirement is met, and if all check items in the users file are matched by additional attributes in the access-request message, the RADIUS server sends an access-accept message to the PortMaster indicating that bob has been successfully authenticated. It also sends authorization information—reply items—about the services bob can access and configuration information about bob's connection.
   - If the password request is not satisfied or if other check items—specified in the RADIUS users file—fail, the RADIUS server sends an access-reject message to the PortMaster indicating that the authentication attempt has failed. The PortMaster terminates bob’s connection attempt.

If third-party software—such as ActivCard or SecurID—is used, the user might be prompted for more information before being accepted or rejected.

- **Authorization**
  
  Authorization controls access to specific services on the network by configuring the user's session. Once a user is authenticated, RADIUS reports to the PortMaster what a user is authorized (permitted) to access. For example, user bob might be authorized to use the Point-to-Point Protocol (PPP) for his connection, be assigned IP address 192.168.200.4, and have to filter his traffic using packet filters std.ppp.in and std.pp.out.

- **Accounting**
  
  RADIUS accounting stores usage information for dial-in users. This information is often used for billing purposes. When the user is authenticated and the session has been configured according to the authorization information, an accounting start record is created. When the user's session is terminated, an accounting stop record is created. See Chapter 8, “Implementing RADIUS Accounting,” for more information.
Ease-of-Use Enhancements

RADIUS 2.1 for UNIX provides the following enhancements to improve RADIUS functionality:

- **Proxy RADIUS**

  Proxy RADIUS enables your RADIUS server to forward authentication requests from a PortMaster or other network access server (NAS) to a remote RADIUS server and to pass the reply back to the NAS. This feature enables cooperating Internet service providers (ISPs) to handle dial-in service requests from each other’s users. Corporate users can easily forward packets from local to remote networks.

  You create a proxy file in the `/etc/raddb` directory on the forwarding server and, if necessary, on the remote server. Each line in the proxy file contains the hostname or IP address of a remote server, the secret shared between the forwarding server and the remote server, and the realm of the remote server in the proxy chain. Each entry can specify the ports the remote server uses for RADIUS authentication and accounting as well as certain keywords to affect server behavior.

  **Note** – Usernames with an embedded at sign (@) are treated as proxy realms.

- **iPass protocol support**

  The iPass protocol enables you to provide global roaming Internet access. To use iPass with RADIUS 2.1, you must do the following:

  b. Add the keyword ipass to the entries of the appropriate remote servers in your `/etc/raddb/proxy` file.
  c. Run `iradiusd` instead of `radiusd`.

  **Note** – If you run iPass and ActivCard or Securid, for configuration information send email to support@livingston.com.

- **ActivCard support**

  RADIUS now supports ActivCard authentication on the following platforms supported by ActivCard 2.1: AIX, HP-UX, Solaris, and Sun-OS. ActivCard authenticates users by means of dynamic passwords generated by a handheld token.
using the public Digital Encryption Standard (DES) algorithm. The RADIUS server can forward all requests specified by the user profiles to the ActivCard server. Perform the following steps to enable user authentication via ActivCard:

a. Install the ActivCard server on either the same host as the RADIUS server or on a different host.

b. Create the file `/etc/raddb/config.aeg` on the host where the RADIUS server resides.

c. Specify Auth-Type = ActivCard as a check item in the user profiles of all ActivCard users.

d. Run the daemon `sradiusd` instead of `radiusd`.

- Accounting signatures required

RADIUS 2.1 strictly complies with RFC 2139. The server discards unsigned accounting packets—packets with invalid request authenticator attributes—and logs an error message. If you want to use RADIUS accounting, you must run RADIUS 2.1 with ComOS 3.3.1 or later. You can use `radiusd -o` to run RADIUS 2.1 if you have noncompliant RADIUS clients. In this case, RADIUS logs unsigned accounting records and flags them with Request-Authenticator = None.

- Virtual ports

You can restrict the number of logins permitted to specified telephone numbers.

a. Install the RADIUS accounting and authentication servers on the same host.

b. Create the file `/etc/raddb/vports` with the format shown in the following sample file:

```
# Called-Station-ID   Number of logins permitted
9255550020            20
9255550021            30
9255550022            25
```

c. Issue the command `radiusd -s` to run the server in single-thread mode.

**Note** – The virtual ports feature does not provide an exact control. Logins that occur before `radiusd` starts running are not considered in the count. Accounting records that are sent to the backup accounting server are not considered in the count. This feature cannot provide simultaneous login limits for individual users because it is based on Called-Station-Id rather than Calling-Station-Id.
• Vendor-specific attributes

RADIUS 2.1 supports RFC-compliant vendor-specific attributes, including the two new Lucent Remote Access attributes introduced in ComOS 3.8: LE-Advice-of-Charge and LE-Terminate-Detail.

The LE-Advice-of-Charge value is a string included in RADIUS accounting stop records generated by ComOS 3.8 or later. This string provides any advice-of-charge information passed along by the telephone company on the ISDN D channel.

The LE-Terminate-Detail value is a string included in RADIUS accounting stop records generated by ComOS 3.8 or later. This string provides a detailed description of the reason the session terminated.

The RADIUS 2.1 dictionary file uses the following syntax to define vendor-specific attributes that conform to RFC 2138:

```#
# Vendor-Specific attributes use the SMI Network Management Private
# Enterprise Code from the "Assigned Numbers" RFC
#
VENDOR Livingston 307
#
# Livingston Vendor-Specific Attributes (requires ComOS 3.8 and RADIUS
s# 2.1)
ATTRIBUTE LE-Terminate-Detail 2 string Livingston
ATTRIBUTE LE-Advice-of-Charge 3 string Livingston
```

• Alternate password file

You can run `radiusd -f Filepath` to specify a password file other than `/etc/passwd`.

• Address binding

You can force the RADIUS server to bind to a specific IP address to listen for requests by running `radiusd -i Ipaddress`. You might find this useful if you are running RADIUS on a multihomed host or a host with a virtual IP address.

**Note** – When using hosts that are multihomed or that have virtual IP addresses, remember RADIUS clients ignore replies that do not originate from the primary or secondary RADIUS server specified on the client.
• Improved messages

The **syslog** message for many kinds of rejected access-requests now includes the Calling-Station-Id—if known—enabling you to track down where the failed login attempts are dialing from. Here is an example **syslog** message:

```
Jul 10 21:10:50 ra radius[14870]:unix_pass: password for "bob" at 5551234 failed
```

Failures are currently logged for unknown users and for failed logins where the user profile included Auth-Type = System.

Other syslog messages are more detailed, including the UDP port and RADIUS message ID for easier tracking. The following example syslog message shows a RADIUS packet being forwarded from UDP port 1093 and ID #139 for the source IP address to UDP port 1645 and ID #17 for the destination IP address:

```
Jul 10 21:10:50 ra radius[14870]:forwarding request from 192.168.96.6/1093.139 to 172.16.3.24/1645.17 for edu.com
```

• Enhanced debugging

You can turn on RADIUS debugging by sending a SIGUSR1 signal to **radiusd**. Sending a SIGUSR2 signal to **radiusd** turns debugging off. The RADIUS server logs a short summary message of **radiusd** activity when either signal is sent and when **radiusd** is exited. See “Turning on the Debug Function” on page A-5 for instructions.

**Feature No Longer Supported**

The obsolete RADPASS feature is no longer available.

**RADIUS Directory Structure**

RADIUS server files are stored in the **raddb** (RADIUS database) directory. The **raddb** directory is typically placed within the **/etc** directory. Lucent Remote Access recommends that you use this default.

The **raddb** directory contains files and subdirectories organized as shown in Figure 1-2 and explained in the list that follows. A dotted line indicates an optional file.
The **users** file stores user profiles, which consist of authentication and authorization information for all users authenticated with RADIUS.

The **dictionary** file contains information used to parse user access requests and generate responses. It lists attribute numbers, names, and the data type expected for attribute values. Attributes appear in client-server communications and accounting records and are used to create user profiles. After you modify this file, you must restart RADIUS to apply your changes.

The **clients** file contains the IP addresses of all RADIUS clients and the secrets shared between the clients and the RADIUS server(s).

The **builddb** utility enables you to convert the **users** file into DBM format for faster look-ups.

The **menus** subdirectory contains individual menu text files. You can optionally use menus to provide users with different login options once the users have been authenticated.

The **proxy** file stores information about the remote servers to which the local server can forward authentication requests. This information includes the hostname or IP address, shared secret, and the remote realm.

The **vports** file contains Called-Station-IDs and the number of logins permitted via each telephone number.

The **config.aeg** file contains the configuration settings that specify how RADIUS connects to an ActivCard server. This optional file is needed only if you are using ActivEngine with RADIUS. See Chapter 6, “Installing and Configuring ActivCard,” for more information.
The RADIUS server uses the User Datagram Protocol (UDP) and the following UDP ports:

- Port 1645 for authentication
- Port 1646 for accounting
- Ports 1650 and 1651 for proxy

If different ports are assigned to these services in the /etc/services file, RADIUS uses those ports in preference to the default ports listed above. You can also specify different UDP ports by using the `radiusd -p portnumber` command on UNIX hosts. Port 1812 is reserved for RADIUS authentication and port 1813 is reserved for RADIUS accounting. See Appendix C, “RADIUS Options,” for more information.

\* Note – PortMaster products use ports 1645, 1646, 1650, and 1651 by default; this is specified by ComOS and cannot be modified in ComOS versions prior to 3.8. If you change the port number as stated above, RADIUS might work with other network access servers (NASs) but cannot authenticate users or gather accounting data for accesses to PortMaster products unless they are running ComOS version 3.8 or later.

Table 1-1 provides a quick overview of the tasks required to install and configure RADIUS.

**Table 1-1  Overview of RADIUS Installation and Configuration Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select a host to use as the RADIUS server.</td>
<td>See “Getting Started” on page 2-1.</td>
</tr>
<tr>
<td>2. Install the RADIUS server software on the host.</td>
<td>• See “Installing RADIUS on a UNIX Host” on page 2-3</td>
</tr>
<tr>
<td>3. Configure client information on the RADIUS server.</td>
<td>See “Modifying the clients File” on page 3-1.</td>
</tr>
</tbody>
</table>
Table 1-1  Overview of RADIUS Installation and Configuration Tasks (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Instructions</th>
</tr>
</thead>
</table>
| 4. Configure the PortMaster as a RADIUS client. | See one of the following:  
• “Configuring the PortMaster Using the Command Line Interface” on page 3-2.”  
• “Configuring the PortMaster Using PMVision” on page 3-4.” |
| 5. Configure user profiles. | See Chapter 4, “Configuring User Information.” |
| 6. You can optionally define menus to enable authenticated users to select different login options. | See Chapter 5, “Configuring RADIUS Menus.” |
| 8. You can optionally install and configure ActivCard. | See Chapter 6, “Installing and Configuring ActivCard.” |
| 7. You can optionally install and configure SecurID. | See Chapter 7, “Installing and Configuring SecurID.” |
| 9. You can optionally install and configure RADIUS accounting. | See Chapter 8, “Implementing RADIUS Accounting.” |
| 10. You can optionally configure RADIUS proxy service. | See Chapter 9, “Configuring RADIUS Proxy Service.” |

1. This installation requires ActivCard software. If you use this software, you must run `sradiusd` rather than `radiusd`. If you use this software with iPass, send email to support@livingston.com.

2. This installation requires ACE/Server and ACE/Client software. If you use this software, you must run `sradiusd` rather than `radiusd`. If you use this software with iPass, send email to support@livingston.com.
Configuring a RADIUS Server

This chapter includes the following topics:

• “Getting Started” on page 2-1
• “Installing RADIUS on a UNIX Host” on page 2-3

Getting Started

Before installing and configuring RADIUS software, select a host or hosts to use as a RADIUS server and determine one or more shared secrets for authentication.

Selecting a RADIUS Server Host

Primary RADIUS Authentication Server. Select or create a host with the following characteristics to use as a RADIUS authentication server:

• Secure physical location
• Root access limited to the security officer or system administrator
• Limited number of user accounts—preferably none
• Basic memory and disk space

Lucent Remote Access suggests the following additional characteristics for the host:

• Inaccessibility from outside your local network
• Absence of public network services such as email, FTP, HTTP, netnews, Telnet, rlogin, and rcp.

Note – RADIUS performance varies with the number of users being authenticated and with other demands on the server. Running public network services or other applications on the server concurrently with RADIUS can consume most of your CPU resources. You can experience a reduction in RADIUS performance—such as access denials or dropped calls—if you have insufficient CPU resources on the host. Lucent Remote Access strongly recommends that you do not run a Web server on the RADIUS
for UNIX server. If you do, a sudden increase in hits on the Web server can adversely affect your RADIUS processes, and you might be subject to malicious attacks by this approach.

**Secondary RADIUS Authentication Server.** Lucent Remote Access recommends the use of a secondary RADIUS server with the same security and performance characteristics as the primary server. The PortMaster always queries the primary RADIUS server first; if the server does not respond, it is queried a second time. Then both the primary and secondary servers are queried up to eight more times at 3-second intervals until one responds or until 3 seconds after the tenth query without a response. At this point, the login attempt fails.

**RADIUS Accounting Server.** If you implement RADIUS accounting, you must also select one or more RADIUS accounting servers. The RADIUS accounting server can be located on the same host as the RADIUS server used for authentication, or on a separate host. See Chapter 8, “Implementing RADIUS Accounting,” for more information.

**Secondary RADIUS Accounting Server.** You can define a secondary accounting server to serve as a backup if the primary server cannot be contacted. The PortMaster always sends accounting packets to the primary RADIUS accounting server first, and retries it once every 45 seconds. If the primary server does not respond within 10 minutes, or if more than 50 accounting packets are waiting to be sent, the PortMaster sends the accounting packets to the secondary RADIUS accounting server. This behavior is subject to change in future releases of ComOS.

### Determining a Shared Secret

Each PortMaster using RADIUS shares an authentication key—called the **shared secret**—with its RADIUS server(s). The shared secret consists of up to 15 printable, nonspace, ASCII characters. The RADIUS server can have a different shared secret with each PortMaster client, or the RADIUS server can have the same shared secret for multiple PortMaster clients. Different, nontrivial shared secrets are recommended for each PortMaster.

You configure the shared secret on each RADIUS server and the PortMaster. It is stored as clear text in the **clients** file on the RADIUS server and in the nonvolatile memory of the PortMaster. See Chapter 3, “Adding a RADIUS Client”, for more information.
Installing RADIUS on a UNIX Host

Use one of the following installation methods:

- Install RADIUS with the `pminstall` utility
- Install RADIUS without `pminstall`.


Installation with pminstall

To install RADIUS using `pminstall`, complete the following steps:

1. **Log in to the selected RADIUS server as root.**
2. **Mount the CD using the instructions in the CD booklet.**

   For the instructions here, an example mount point of `/cdrom` is used. Change this as needed for your operating system and mount point.

3. **Enter the following command to install the PortMaster software:**

   `/cdrom/lei/unix/setup`

4. **Enter the `/usr/portmaster/pminstall` command at the UNIX prompt.**

   The following list of choices appears:

   `% /usr/portmaster/pminstall`

   1. PortMaster Internet Address Setup
   2. Host Installation
   3. PortMaster Upgrade
   4. Host Upgrade
   5. Install RADIUS
   6. Install ChoiceNet
   7. Exit

   Please select an option from above:
5. **Choose the** Install RADIUS option to install all RADIUS files.

The server prompts you for directory names:

- Database installation directory (/etc/raddb):
- RADIUS accounting log directory (/usr/adm/radacct):
- Directory to install radiusd in (/etc):

6. **Provide directory information for RADIUS files by one of the following methods:**

   - Select the default directory (shown in parentheses) by pressing the **Return** or **Enter** key.
   - Enter the appropriate directory.

7. **When RADIUS installation is complete, select the** Exit option to quit pminstall.

8. **Enter the following command to start the RADIUS server:**

   /etc/radiusd

   The **radiusd** process spawns RADIUS accounting as a child process.

   **Note** – **radiusd** is a standalone process; it cannot be run from /etc/inetd.conf.

   For usage of options with the **radiusd** command, see Appendix C, “RADIUS Options.” Running **radiusd -b** enables you to use cached user profiles. Lucent Remote Access recommends caching user profiles when the **users** file contains more than 500 users. See “Configuring Database Caching of User Profiles” on page 4-40 for more information.

9. **Go to Chapter 3, “Adding a RADIUS Client.”**

**Installation without pminstall**

To install RADIUS without **pminstall**, complete the following steps:

1. **If you are running the Network Information Service (NIS) or NIS+, add the following lines to the services NIS map on your NIS master and push the maps.**

   radius 1645/udp radiusd
   radacct 1646/udp
2. Log in to the selected RADIUS server as root.

3. Mount the CD using the instructions in the CD booklet.

   For the instructions here, an example mount point of /cdrom is used. Change this as needed for your operating system and mount point.

Note – RADIUS server version 1.16 required you to specify the RADIUS ports by adding the lines shown in Step 1 to the /etc/services file. RADIUS server versions 2.0 and higher use ports 1645 and 1646 by default and do not require modification of /etc/services. However, if nondefault ports are specified in the /etc/services file, the values assigned override the default values. You can use radiusd with the -p option to override both the default RADIUS values and any values specified in the /etc/services file.

4. As root, enter the following commands on the RADIUS server:

   umask 022
   mkdir -p /etc/raddb /usr/adm/radacct
   chmod 700 /etc/raddb /usr/adm/radacct

   The commands in this example create two directories, raddb and radacct. All RADIUS files (except the radiusd executable) are stored in the /etc/raddb directory. The radacct directory is used to store RADIUS accounting logs.

   The umask and chmod commands affect the raddb and radacct directory permissions; root access is required for read, write, and execute privileges.

Caution – If you are upgrading from an existing installation of RADIUS, save the files in /etc/raddb before performing Step 5.

5. Copy all files in /cdrom/lei/unix/radius/raddb to the /etc/raddb directory:

   cp -r /cdrom/lei/unix/radius/raddb/* /etc/raddb

   In RADIUS for UNIX version 2.0 or later, the raddb directory contains three files—users, clients, and dictionary—and the menus directory.
6. **Copy the `radiusd` file to the `/etc` directory** (or if you prefer, to another directory such as `/usr/sbin`). Replace `platform` with the name of the CD-ROM directory for your operating system—for example, `sun4_4.1`:

```
cp /cdrom/lei/unix/platform/radiusd /etc/radiusd
```

7. **Copy the `builddbm` utility to `/etc/raddb/builddbm`**. Replace `platform` with the name of the CD-ROM directory for your operating system—for example, `sun4_4.1`:

```
cp /cdrom/lei/unix/platform/builddbm /etc/raddb/builddbm
```

8. **If you are using proxy service, create the `/etc/raddb/proxy` file**.

   See Chapter 9, “Configuring RADIUS Proxy Service,” for information on configuring proxy service.

9. **Use the `radiusd` command to start RADIUS**:  
    
    ```
    /etc/radiusd
    ```

    The `radiusd` daemon spawns the RADIUS accounting server as a child process. For more information about RADIUS accounting, see Chapter 8.

    If you are using iPass, run `iradiusd` instead of `radiusd`. If you are using ActivCard authentication, run `sradiusd` instead of `radiusd`.

**Note – `radiusd` is a standalone process; it cannot be run from `/etc/inetd.conf`.**

For usage of options with the `radiusd` command, see Appendix C, “RADIUS Options.” Running `radiusd -b` enables you to use cached user profiles. Lucent Remote Access recommends caching user profiles when the `users` file contains more than 500 users. See “Configuring Database Caching of User Profiles” on page 4-40 for more information.
10. **To start the** radiusd **daemon each time the operating system is booted, place following script in your system start-up scripts:**

   ```
   # Start RADIUS
   #
   if [ -x /etc/radiusd ]; then
     echo "RADIUS"
     /etc/radiusd #add flags here
   fi
   ```

   Consult your UNIX system documentation for more information. Example start-up scripts are `/etc/rc.local` for SunOS 4.1.4, `/etc/rc2.d/S99radiusd` for Solaris 2.5.1, or `/etc/rc.d/rc.local` for Linux.

   **Note** – **radiusd** does not need to be restarted each time the clients or users files are modified. This daemon needs to be restarted only when the dictionary file is modified.

11. **Go to Chapter 3, “Adding a RADIUS Client.”**
Adding a RADIUS Client

This chapter includes the following topics:

• “Modifying the clients File” on page 3-1
• “Configuring the PortMaster Using the Command Line Interface” on page 3-2
• “Configuring the PortMaster Using PMVision” on page 3-4

This chapter describes adding a PortMaster as a RADIUS client. There are two steps to adding a RADIUS client:

1. **Modify the clients file to add the PortMaster and shared secret.**
2. **Configure the following on the PortMaster and save the configuration changes.**
   - Security enabled on all ports
   - IP addresses of the primary and optional alternate RADIUS authentication servers; optionally configure an authentication port number different from the default
   - IP addresses of the primary and optional alternate RADIUS accounting servers, if accounting is to be performed; optionally configure an accounting port number different from the default
   - RADIUS shared secret

You configure RADIUS clients using the PortMaster command line interface (see “Configuring the PortMaster Using the Command Line Interface” on page 3-2) or using a graphical user interface (GUI) (see “Configuring the PortMaster Using PMVision” on page 3-4).

**Modifying the clients File**

The `clients` file is a flat text file installed on the RADIUS server. The `clients` file stores information about RADIUS clients, including each client’s name or IP address and its shared secret. Use any text editor to edit the `/etc/raddb/clients` file.
1. **Verify that only root users have read and write access to the clients file.**

   The clients file contains the shared secrets for the RADIUS clients, and this information must be protected from unauthorized access.

   The permissions on a UNIX host look like this:

   -rw------- 1 root daemon 802 Jul 15 00:21 clients

2. **To add a client, enter the client's name or IP address and the shared secret. To add a comment line, start the line with the number sign (#).**

   Shared secrets must consist of 15 or fewer printable, nonspace, ASCII characters. There is no limit to the number of clients that you can add to this file.

   Here are some examples of client names and shared secrets:

<table>
<thead>
<tr>
<th>#Client Name</th>
<th>Shared Secret</th>
</tr>
</thead>
<tbody>
<tr>
<td>portmaster1</td>
<td>wP40cQ0</td>
</tr>
<tr>
<td>portmaster2</td>
<td>A3X445A</td>
</tr>
<tr>
<td>192.168.1.2</td>
<td>wer369st</td>
</tr>
</tbody>
</table>

   **Note** – Lucent Remote Access recommends that you use IP addresses to avoid the DNS lookup time entailed by using client names and possible incorrect name translation.

3. **Go to one of the following sections to configure the PortMaster as a RADIUS client:**
   - “Configuring the PortMaster Using the Command Line Interface” in the next section
   - “Configuring the PortMaster Using PMVision” on page 3-4

---

**Configuring the PortMaster Using the Command Line Interface**

To configure the PortMaster using the command line interface, complete the following steps.

1. **Enable port security on all ports using the following command:**

   Command> set all security on
The PortMaster tries to authenticate each user attempting to log in to a port by looking up the user in its user table. RADIUS authenticates users when port security is enabled and the user is not found in the user table. When port security is disabled and the user is not found in the PortMaster user table, RADIUS is not used and the user is passed through to the login host without further authentication.

2. **Enter the IP address, and optionally the authentication port number, of the primary RADIUS server using the following command:**

   Command> `set authentic Ipaddress [Uport]`

   The default RADIUS authentication port, 1645, is used if you specify a port number of 0 or do not specify a port number.

3. **You can optionally specify a secondary (alternate) RADIUS server:**

   Command> `set alternate Ipaddress [Uport]`

   The PortMaster consults the primary RADIUS server first. If the server does not respond within 3 seconds, it is queried a second time; then both servers are queried up to eight additional times at 3-second intervals.

4. **To log activity using RADIUS accounting, enter the IP address, and optionally the accounting port number, of the primary accounting server:**

   Command> `set accounting Ipaddress [Uport]`

   The default RADIUS accounting port, 1646, is used if you specify a port number of 0 or do not specify a port number.

5. **You can optionally specify a secondary (alternate) accounting server:**

   Command> `set accounting 2 Ipaddress [Uport]`

   Lucent Remote Access recommends the use of a secondary RADIUS accounting server. The PortMaster always sends accounting packets to the primary RADIUS accounting server first, and retries it once every 45 seconds. If the primary server does not respond within 10 minutes, or if there are more than 50 accounting packets waiting to be sent, the PortMaster sends the accounting packets to the secondary RADIUS accounting server.

6. **Enter the secret shared by the PortMaster and RADIUS server using the `set secret` command:**

   Command> `set secret String`
This is the same shared secret entered in the clients file on the RADIUS server (see page 3-1).

Note – The shared secret is a string of up to 15 printable, nonspace, ASCII characters. If a secret longer than 15 characters is specified, an error message is displayed. Secrets in the clients file and configured on the PortMaster are case-sensitive and must match exactly.

7. Save your changes using the save all command; then reset all ports:

Command> save all
Command> reset all

Caution – Resetting all ports disconnects any user sessions in progress. Resetting is only necessary when changes have been made to serial ports.

8. Continue to Chapter 4. “Configuring User Information.”

Configuring the PortMaster Using PMVision

You can use the PMVision application, a Java GUI, instead of the command line, to configure your PortMaster clients. PMVision provides all configuration options available through the older PMconsole interface.

Perform the following steps to configure a PortMaster as a client. Refer to the PMVision online help for more information on using PMVision. After configuring the client using PMVision, go to Chapter 4, “Configuring User Information.”

1. From PMVision, select PortMaster>Configure>RADIUS to display the RADIUS configuration panel.
2. Select the PortMaster you want to configure as a RADIUS client.
3. Enter the IP address of the primary authentication server.
4. Optionally, enter the IP address of a secondary authentication server.
5. Enter the IP address of the primary accounting server.
6. Optionally, enter the IP address of a secondary accounting server.
7. Enter the shared secret.

For security, the shared secret is not displayed in the field.
8. **Click** Save....

Figure 3-1 shows a close-up of the RADIUS panel with saved configuration settings.

*Figure 3-1  Detail View of RADIUS Configuration on PMVision*

<table>
<thead>
<tr>
<th>PortMaster — Configure — RADIUS</th>
<th>192.168.113.22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Authentication Server</td>
<td>192</td>
</tr>
<tr>
<td>Alternate Authentication Server</td>
<td>192</td>
</tr>
<tr>
<td>Primary Accounting Server</td>
<td>192</td>
</tr>
<tr>
<td>Alternate Accounting Server</td>
<td>192</td>
</tr>
</tbody>
</table>

**Note** – The PMVision display varies depending on the version of ComOS running on the selected PortMaster. For example, if the selected PortMaster is running ComOS 4.0 or later, the RADIUS configuration panel enables you to select the ports used by the authentication and accounting servers.
This chapter includes the following topics:

- “User Profile Format” on page 4-2
- “Matching User Profiles” on page 4-3
- “Editing User Profiles” on page 4-4
- “Default User Profiles” on page 4-4
- “Check Items” on page 4-6
- “Reply Items” on page 4-19
- “Using RADIUS with PAP and CHAP” on page 4-38
- “Configuring Database Caching of User Profiles” on page 4-40
- “Example PPP User Profile” on page 4-40

The RADIUS users file is a flat text file on the RADIUS server. The users file stores authentication and authorization information for all users authenticated with RADIUS. Each user must be represented by a profile that consists of three parts: the **username**, a list of **check items**, and a list of **reply items**. Figure 4-1 displays an example.

### Figure 4-1  User Profile

<table>
<thead>
<tr>
<th>username</th>
<th>check Item</th>
<th>1st reply item</th>
<th>2nd reply item</th>
<th>final reply item</th>
</tr>
</thead>
<tbody>
<tr>
<td>bob</td>
<td>Password = &quot;ge55ep&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service-Type = Framed-User,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-Protocol = PPP,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-Routing = None,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-IP-Address = 255.255.255.254,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-IP-Netmask = 255.255.255.255,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-Compression = Van-Jacobson-TCP-IP,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filter-Id = &quot;firewall&quot;,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Framed-MTU = 1500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note – Lucent Remote Access recommends that you create a single user profile and test RADIUS authentication and authorization for that user. If the test is successful, create and test profiles representative of other user types before proceeding to create all your user profiles.

User Profile Format

User profiles must be separated from each other by an empty line. The first line of a user profile consists of the username followed by the check items. The username is separated from the check items by spaces or tabs. This first line must not exceed 255 characters. All subsequent lines of the profile are individual reply items. Each reply item line must begin with a space or tab. Each reply item, except for the final line in the profile, must end with a comma.

You can add comments to the users file by beginning comment lines with a number sign (#).

Caution – Do not place comments within a user profile. Comments in a user profile prevent any reply item following the comment from being processed and sent to the client. Place comments either before or after the user profile.

The contents of each user profile are case-sensitive. Definitions for all attributes and values are in the dictionary file and can be viewed with any text editor.

Caution – Modifying the contents of the dictionary file incorrectly can cause RADIUS to fail to authenticate users correctly or cause other problems.

See “Default User Profiles” on page 4-4 for information on the special profile, DEFAULT. Several common user profiles are listed in “Example PPP User Profile” on page 4-40. All check items are summarized in Table 4-1 on page 4-7. All reply items are summarized in Table 4-2 on page 4-19. Attributes and values used to create user profiles are defined in the dictionary.

Username. The username is the first part of each user profile and must start in the first column. Usernames consist of up to 63 printable, nonspace, ASCII characters. If ActivCard, SecurID, or a system password file is used for authentication, the username must conform to any limitations imposed on the username by the host.

Caution – Do not use white space within a username. In RADIUS 2.1, access-requests are rejected if the username contains a space or tab. If a user enters a username with trailing spaces or tabs, the access-request is rejected.
Matching User Profiles

**Check Items.** Check items are listed on the first line of a user profile, following the username and separated from it by white space. The line in the user profile that contains the username and check items must not exceed 255 characters. Check items must be separated by commas. Do not place a comma after the final check item. For an access-request (see “How RADIUS Works” on page 1-2) to succeed, all check items in the user profile must be satisfied by information from the access-request or by related information from the local system, such as group membership in the access-request.

In Figure 4-1, *bob*’s password is the only check item. To successfully authenticate *bob*, the RADIUS server must receive this password in *bob*’s access-request.

The following example shows the first line of a user profile. To successfully authenticate *ann*, the RADIUS server must receive the specified password, *ann* must be logging in on port 23 of the NAS, and she must be a member of the engineer group.

```
ann Password = "8f4kv$s", NAS-Port = 23, Group = "engineer"
```

**Note** – If no check items are included in the user profile, the user is rejected.

**Reply Items.** Reply items are placed one per line. Each line begins with white space. Each line ends with a comma, except for the final reply item. Reply items give the PortMaster authorization information about the user’s connection—for example, whether PPP or SLIP is used or whether the user’s IP address is negotiated. In Figure 4-1, Framed-Protocol is a reply item. The value of Framed-Protocol is PPP, indicating that *bob* uses PPP for his connection.

If all check items in the user profile are satisfied by the access-request, the RADIUS server sends the reply items to the PortMaster to configure the connection.

**Matching User Profiles**

When a user logs in, the RADIUS server searches the users file for a matching profile. The following components of a profile must match the access-request for authentication to occur:

1. username
2. password check item
3. other check items
The username matches if any of the following conditions are met:

- The username in the profile is identical to the login name in the access-request.
- The username in the profile is DEFAULT or DEFAULT#, where # is any integer. See “Default User Profiles” on page 4-4 for more information.

The password matches if it is identical to that entered by the user. The password can be stored locally in the profile or remotely in a separate file. If you use an additional level of password security, you can specify the additional password authentication step in the profile.

All check items specified in a profile also must be present in the access-request packet or satisfied by local system information, for a match to occur.

**Editing User Profiles**

User profiles are maintained in the users file. On a UNIX host, use any text editor to edit the /etc/raddb/users file.

**Default User Profiles**

When the RADIUS server receives a login name from a PortMaster, the server scans the users file for a matching username, starting from the top of the file. If a match is located, RADIUS attempts to authenticate the user with the information in that user profile. If a matching user profile is not found during the scan, but a DEFAULT profile is located, RADIUS attempts to use the DEFAULT profile for authentication. The DEFAULT profile is typically used when the Auth-Type is System, SecurID, or ActivCard.

**Caution** – You must place DEFAULT profiles at the end of the users file. RADIUS stops scanning profiles when a matching DEFAULT profile is found and ignores any user profiles located after a DEFAULT user profile.

In the following example, user bob’s password is stored in a system password file. When he attempts to connect to the network, RADIUS scans the users file to determine if it contains a matching user profile. If a matching profile is not found before the DEFAULT profile is found, the DEFAULT profile is used. Because the DEFAULT profile includes `Framed-Protocol = PPP` as a reply item, PPP is used for bob’s connection.
Default User Profiles

[Configuring User Information 4-5]

DEFAULT

- Auth-Type = System
- Service-Type = Framed-User,
- Framed-Routing = None,
- Framed-Protocol = PPP,
- Framed-IP-Address = 255.255.255.254,
- Framed-MTU = 1500

RADIUS for UNIX 2.0 and later versions permit multiple DEFAULT user profiles. In place of a username, the first line of DEFAULT profiles start as follows:

- **DEFAULT**, with all capital letters. You can use this for multiple DEFAULT profiles if the profiles include different check items.

- If there are multiple default profiles you can append a number to the end of the profile name—for example, DEFAULT1, DEFAULT2, and so on:

  DEFAULT1
  - Auth-Type = System, Called-Station-Id = 9255551234
  - (reply items)

  DEFAULT2
  - Auth-Type = System, Called-Station-Id = 9255554971
  - (reply items)

In the following example, the **Prefix** and **Suffix** check items distinguish between the DEFAULT profiles. See “Username Prefixes and Suffixes” on page 4-12 for more information on using prefixes and suffixes. When users add the prefix or suffix to their username, the RADIUS server matches them to the corresponding DEFAULT profile.

In the example shown on the next page, assume that user **bob**'s password is stored in a system password file and that there is no profile with a username of **bob** in the RADIUS **users** file.

Suppose **bob** logs in as **Pbob**. When the first DEFAULT profile is found by RADIUS, the server strips the initial **P** from **Pbob** and looks in the system password file for a password associated with **bob**. If the password matches the one user **bob** entered, then he is authenticated as a PPP user.
Check Items

**DEFAULT**

Auth-Type = System, Prefix = "P"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Framed-IP-Address = 255.255.255.254,
Framed-MTU = 1500

**DEFAULT**

Auth-Type = System, Suffix = "%S"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = SLIP,
Framed-IP-Address = 255.255.255.254,
Framed-Compression = None,
Framed-MTU = 1006

**DEFAULT**

Auth-Type = System
Service-Type = Login-User,
Login-IP-Host = 172.16.1.4
Login-Service = Telnet

If *bob* logs in as *bob%S*, the first DEFAULT profile is not used because there is no initial *P* present in the login name. When the second DEFAULT profile is found by RADIUS, the server strips the ending *%S* from *bob%S* and looks in the system password file for a password associated with *bob*. If the password matches the one user *bob* entered, then he is authenticated as a SLIP user.

If *bob* logs in as *bob*, the first two DEFAULT profiles are not used because the Prefix and Suffix check items do not match. When the third DEFAULT profile is found by RADIUS, the server looks in the system password file for a password associated with *bob*. If the password matches the one user *bob* entered, then he is authenticated as a Telnet user.

**Check Items**

Check items are used to authenticate the user. Table 4-1 describes all check items that can be used in RADIUS user profiles. Called-Station-Id, Calling-Station-Id, and Connect-Rate can be used as check items only if the RADIUS client is capable of sending them in an access-request. Called-Station-Id and Calling-Station-Id are supported over an ISDN Primary Rate Interface (PRI) on a PortMaster 3 running ComOS 3.7 or later and on a PortMaster 4 running ComOS 4.0 or later, and over an ISDN Basic Rate Interface (BRI) on other PortMaster products running ComOS 3.7 or later. Connect-Rate is supported only on a PortMaster 4 running ComOS 4.0 or later or a PortMaster 3 running ComOS 3.7 or later.
Check Items

**Note** – Although it is not considered a check item, be sure that the username appears as the first item on the first, or check item, line of the user profile.

<table>
<thead>
<tr>
<th>Item</th>
<th>Options</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth-Type</td>
<td>Local</td>
<td>User’s password is stored in the RADIUS users file. Default.</td>
</tr>
<tr>
<td></td>
<td>System</td>
<td>User’s password is stored in a system password file.</td>
</tr>
<tr>
<td></td>
<td>ActivCard</td>
<td>User is authenticated via ActivEngine software.</td>
</tr>
<tr>
<td></td>
<td>SecurID</td>
<td>User is authenticated via ACE/Server software.</td>
</tr>
<tr>
<td></td>
<td>Reject</td>
<td>User always fails authentication.</td>
</tr>
<tr>
<td>Called-Station-Id</td>
<td>String of numerals</td>
<td>Telephone number called by user. Available in ComOS 3.7 for ISDN BRI and PortMaster 3 ISDN PRI. Available in ComOS 4.0 for PortMaster 4 ISDN PRI.</td>
</tr>
<tr>
<td>Calling-Station-Id</td>
<td>String of numerals</td>
<td>Telephone number user is calling from. Available in ComOS 3.7 for ISDN BRI and PortMaster 3 ISDN PRI. Available in ComOS 4.0 for PortMaster 4 ISDN PRI.</td>
</tr>
<tr>
<td>Connect-Rate</td>
<td>String of numerals</td>
<td>Maximum connection rate permitted, in bps. Available for the PortMaster 3 running ComOS 3.7 or later or the PortMaster 4 running ComOS 4.0 or later.</td>
</tr>
<tr>
<td>Crypt-Password</td>
<td>User’s password</td>
<td>User’s password is stored in UNIX crypt format. CHAP authentication attempts fail if Crypt-Password is used, even if the password is correct.</td>
</tr>
</tbody>
</table>
### Table 4-1  User Profile Check Items (Continued)

<table>
<thead>
<tr>
<th><strong>Item</strong></th>
<th><strong>Options</strong></th>
<th><strong>Explanation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration</td>
<td>Must be specified in “Mmm dd yyyy” format</td>
<td>Date that user’s password expires.</td>
</tr>
<tr>
<td>Framed-Protocol</td>
<td>PPP</td>
<td>PPP is used for the connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can also be used as a reply item.</td>
</tr>
<tr>
<td>Group</td>
<td>String of characters in double quotation marks (&quot; &quot; )</td>
<td>Groups that user belongs to.</td>
</tr>
<tr>
<td>NAS-IP-Address</td>
<td>IP address</td>
<td>PortMaster IP address.</td>
</tr>
<tr>
<td>NAS-Port</td>
<td>Number</td>
<td>The PortMaster port number that the user is dialed in to (for example, NAS-Port = S2).</td>
</tr>
<tr>
<td>NAS-Port-Type</td>
<td>ISDN</td>
<td>ISDN port.</td>
</tr>
<tr>
<td></td>
<td>Async</td>
<td>Asynchronous port.</td>
</tr>
<tr>
<td></td>
<td>Sync</td>
<td>Synchronous port.</td>
</tr>
<tr>
<td></td>
<td>ISDN-V120</td>
<td>ISDN in V.120 mode.</td>
</tr>
<tr>
<td></td>
<td>ISDN-V110</td>
<td>ISDN in V.110 mode.</td>
</tr>
<tr>
<td>Password</td>
<td>String of characters in double quotation marks (&quot; &quot; )</td>
<td>User’s password.</td>
</tr>
<tr>
<td>Prefix</td>
<td>String of characters in double quotation marks (&quot; &quot; )</td>
<td>Removed from beginning of username before checking password.</td>
</tr>
<tr>
<td>Service-Type</td>
<td>Call-Check</td>
<td>Authenticates the user at the point of entry on a NAS before answering the call. The NAS must support an ISDN Primary Rate Interface (PRI). You must also configure the call-check feature on the NAS.</td>
</tr>
<tr>
<td></td>
<td>Framed-User</td>
<td>User uses PPP or SLIP for the connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can also be used as a reply item.</td>
</tr>
</tbody>
</table>
Passwords

If you are using ComOS 3.5 or later, the user’s local password can be up to 48 printable, non-space, ASCII characters. If you are using an earlier version of ComOS, the password must not exceed 16 characters. The password check item must be contained within double quotation marks. In addition to the password itself, you can specify two different password characteristics in a user profile: the password’s location and its expiration date.

Password Locations

Use the Auth-Type check item to specify the type of authentication to use for a particular user. Auth-Type can be set to one of the following: Local, System, Reject, ActivCard, or SecurID. If this check item is omitted from the user profile, Local is used.

- Local

To indicate that a user’s password is stored in the RADIUS users file, use the Local Auth-Type.

To set the user’s password, use the Password check item as shown in the following example line from a user profile:

```
bob Auth-Type = Local, Password = "ge55ep"
```

Note – If no Auth-Type is specified in a profile, then the RADIUS server assumes Auth-Type = Local. Typically, you include the Password check item. If you do not include the Password item—that is, if no check items are specified in the profile—then the user is rejected.
Check Items

- **System**
  
  To indicate that a user’s password is stored in a system password file, use the System Auth-Type:
  
  ```
  bob Auth-Type = System
  ```
  
  System can be a password file in UNIX, such as `/etc/passwd`, `/etc/shadow`, or a password map in NIS or NIS+. When the RADIUS server receives a username-password pair from the client and the Auth-Type = System, the server queries the operating system to determine if there is a matching username-password pair.
  
  The System Auth-Type is equivalent to the `Password = “UNIX”` check item in RADIUS server version 1.16, which is also permitted in RADIUS server versions 2.0 and 2.1 for backward compatibility:
  
  ```
  bob Password = "UNIX"
  ```

- **ActivCard**
  
  To specify that the user’s password is to be authenticated by ActivCard, use the ActivCard Auth-Type:
  
  ```
  bob Auth-Type = ActivCard
  ```
  

- **SecurID**
  
  To specify that the user’s password is to be authenticated by a SecurID ACE/Server, use the SecurID Auth-Type:
  
  ```
  bob Auth-Type = SecurID
  ```
  
  To use a SecurID passcode, the RADIUS server must be configured as an ACE/Client and you must have an ACE/Server configured—either on the same or a different host. See Chapter 7, “Installing and Configuring SecurID,” for instructions.

- **Reject**
  
  To reject the user’s authentication attempt without having to delete the user profile from the `users` file, use the Reject Auth-Type:
  
  ```
  bob Auth-Type = Reject
  ```
  
  The Reject Auth-Type feature enables you to disable the user account temporarily.
Check Items

Note – The RADIUS server rejects any access-request with an unknown Auth-Type and logs an error message.

Password Expiration Date

To disable logins after a particular date, complete the following steps:

1. **Specify the date of expiration using the Expiration check item.**
   
   The date must be specified in “Mmm dd yyyy” format, as in the following example:
   
   ```
   bob Password = "ge55gep", Expiration = "Dec 04 1996"
   ```

2. **Edit the Password-Expiration and Password-Warning values in the dictionary to meet your security needs.**
   
   For example
   
   ```
   VALUE Server-Config Password-Expiration 30  
   VALUE Server-Config Password-Warning 5
   ```

   The first parameter, Password-Expiration, turns on expiration if the value is greater than 0 (zero), but has no other effect.

   Password-Warning controls when users are notified that their accounts are about to expire. In the example above, users receive warning messages when they log in, starting 5 days before their password expiration date.

   You can turn account expiration off by making these two lines in the RADIUS dictionary into comments.

   Note – A mechanism to permit users to change their passwords is outside the scope of RADIUS.

3. **If you modified the dictionary file, stop and restart the radiusd daemon.**

Password Encryption

- Use the Crypt-Password check item if you store the user’s password in UNIX crypt format:

  ```
  bob Crypt-Password = "ijFYNcSNctBYg"
  ```
The PortMaster RADIUS client encrypts the password that the user enters at login and sends the encrypted password to the RADIUS server. The server compares this password with the encrypted password stored in one of the following locations:

- The user profile in `/etc/raddb/users`, if Auth-Type = Local
- A system password file, such as `/etc/passwd` or `/etc/shadow`, if Auth-Type = System

The encrypted password in this example corresponds to a password of `abcdefgh`. You must use the UNIX `crypt` command to encrypt user passwords to store in the user profile. The UNIX `passwd` command encrypts passwords and stores them in the system password file.

**Note** – Crypt-Password can be used with scripted logins or with PAP. It cannot be used with CHAP.

### Username Prefixes and Suffixes

Use the Prefix and Suffix check items to allow a network user to access multiple services by adding a series of characters to the beginning or end of his username. Prefix and suffix strings must consist of 16 or fewer printable, nonspace, ASCII characters. The prefix and suffix strings must be contained within double quotation marks.

When a user logs in, the RADIUS server searches through the `users` file for a profile that matches the login. If a profile has a Prefix or Suffix check item, the server strips away the specified prefix or suffix character(s) from the login name before checking the password. If the server does not find a profile that matches the login, the RADIUS server tries to match the login against a DEFAULT user profile.

Consider the following example `users` file:

```
Pliesel  Auth-Type = System, Prefix = "P"
        Framed-Protocol = PPP,
        Service-Type = Framed-User,
        Framed-Routing = None,
        Framed-IP-Address = 255.255.255.254,
        Filter-Id = "firewall",
        Framed-MTU = 1500
```
### DEFAULT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth-Type</td>
<td>System</td>
</tr>
<tr>
<td>Suffix</td>
<td>&quot;%slip&quot;</td>
</tr>
<tr>
<td>Framed-Protocol</td>
<td>SLIP</td>
</tr>
<tr>
<td>Service-Type</td>
<td>Framed-User</td>
</tr>
<tr>
<td>Framed-Routing</td>
<td>None</td>
</tr>
<tr>
<td>Framed-IP-Address</td>
<td>255.255.255.254</td>
</tr>
<tr>
<td>Filter-Id</td>
<td>&quot;firewall&quot;</td>
</tr>
<tr>
<td>Framed-MTU</td>
<td>1500</td>
</tr>
</tbody>
</table>

In this example, *liesel's* username and password are stored in a system password file. If *liesel* specifies a username of **Pliesel** when attempting to connect to the PortMaster, the RADIUS server looks up the username. When the profile for **Pliesel** is found, the Prefix check item matches because the login name begins with a *P*. This cues the server to strip away the specified prefix character and check the system password file for *liesel's* password. If a password match is found, **Pliesel** is connected as a PPP user.

If *liesel* specifies a username of **liesel%slip** when she logs in, the RADIUS server finds no match until it gets to the DEFAULT entry. The Suffix check item matches because the login name ends with *%slip*. This cues the server to strip away the specified suffix characters and check the system password file for *liesel's* password. The server next checks the system password for *liesel*. If a password match is found for *liesel*, **liesel%slip** is connected as a SLIP user.

In the next example, the **users** file has no provision for username prefixes:

<table>
<thead>
<tr>
<th>Username</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>liesel</em></td>
<td>Auth-Type</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>Framed-Protocol</td>
<td>PPP</td>
</tr>
<tr>
<td></td>
<td>Service-Type</td>
<td>Framed-User</td>
</tr>
<tr>
<td></td>
<td>Framed-Routing</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>Framed-IP-Address</td>
<td>255.255.255.254</td>
</tr>
<tr>
<td></td>
<td>Filter-Id</td>
<td>&quot;firewall&quot;</td>
</tr>
<tr>
<td></td>
<td>Framed-MTU</td>
<td>1500</td>
</tr>
</tbody>
</table>

| DEFAULT   | Auth-Type              | System                                     |
|           | Suffix                 | "%slip"                                   |
|           | Framed-Protocol        | SLIP                                        |
|           | Service-Type           | Framed-User                                 |
|           | Framed-Routing         | None                                       |
|           | Framed-IP-Address      | 255.255.255.254                           |
|           | Filter-Id              | "firewall"                                 |
|           | Framed-MTU             | 1500                                       |
Again, liesel’s username and password are stored in a system password file. Suppose liesel logs in as Pliesel. There are no profiles with a username of Pliesel or a Prefix check item specifying P. The RADIUS server cannot find a matching user profile and rejects the connection attempt.

Prefixes and suffixes are most useful when defined in a DEFAULT user profile. See “Default User Profiles” on page 4-4 for information on using prefixes and suffixes in a DEFAULT profile.

**Called-Station-Id**

You can use the number that the user is calling as a check item. Suppose user mario calls in using your toll-free access number, 1-800-555-4973. With the user profile shown in the following example, mario fails authentication. He can only be authenticated if he calls in to 510-555-1234.

```
mario Password = "ge55gep", Called-Station-Id = "5105551234"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP
```

**Note** – Use of the Called-Station-Id check item requires that the Called-Station-Id attribute be included in the access-request. A PortMaster 4 running ComOS 4.0 or later and a PortMaster 3 running ComOS 3.7 or later send this attribute—if it is available—to the RADIUS server as part of the access-request. Other PortMaster products running ComOS 3.7 or later send this information as part of the access-request over ISDN Basic Rate Interfaces (BRIs). If the Called-Station-Id attribute is not sent, the Called-Station-Id check item fails to match and the user is rejected.

**Calling-Station-Id**

You can use the number that the user is calling from as a check item. In the following example, cissy is authenticated if she calls from 209-555-5678. If she calls from any other number, she fails authentication.

```
cissy Password = "ge55gep", Calling-Station-Id = "2095555678"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP
```
Note – Use of the Calling-Station-Id check item requires that the Calling-Station-Id attribute be included in the access-request. A PortMaster 4 running ComOS 4.0 or later and a PortMaster 3 running ComOS 3.7 or later send this attribute—if it is available—to the RADIUS server as part of the access-request. Other PortMaster products running ComOS 3.7 or later send this information as part of the access-request if the user is connecting to an ISDN BRI port. If the Calling-Station-Id attribute is not sent, the Called-Station-Id check item fails to match and the user is rejected.

Client Information

Use the NAS-IP-Address check item to specify the IP address of a particular PortMaster. When this setting is used as a check item in a user profile, the user must be attempting to start a connection on the specified PortMaster for the connection to succeed.

Use the NAS-Port check item to specify a particular PortMaster port. To be successfully authenticated, the user must be attempting to log in to this port:

bob Password = "ge55gep", NAS-Port = 23

Use the NAS-Port-Type check item to specify the type of port. Options for the NAS-Port-Type are as follows: Analog, Async, Sync, ISDN, ISDN-V120, or ISDN-V110. The PortMaster must run ComOS release 3.3.1 or later to support NAS-Port-Type.

The following example displays a user profile containing the NAS-IP-Address and NAS-Port-Type settings:

bob Password = "ge55gep", NAS-IP-Address = 192.168.1.54, NAS-Port-Type = ISDN Service-Type = Framed-User, Framed-Routing = None, Framed-Protocol = PPP

Connect-Rate

The Connect-Rate check item can be used with PortMaster 3 and PortMaster 4 clients. Use this to specify the maximum connection rate permitted for a user.

In the following example, with a connection rate of 28800bps, babar fails authentication if he attempts to connect to the PortMaster 3 at a higher rate—for example, 33600bps or 56000bps:
Check Items

babar  
Auth-Type = System, Connect-Rate = 28800  
Service-Type = Framed-User,  
Framed-Routing = None,  
Framed-Protocol = PPP

You can use this check item to restrict access to users of low-speed modems. In the following example, any user dialing in with a connection rate less than or equal to 14399 bps is rejected:

DEFAULT  
Auth-Type = Reject, Connect-Rate = 14399  
Service-Type = Framed-User

Note – Use of the Connect-Rate check item requires that the Connect-Info attribute be included in the access-request. The PortMaster 4 running ComOS 4.0 or later and the PortMaster 3 running ComOS 3.7 or later send the Connect-Info attribute as part of the access-request. If Connect-Info is not sent, the Connect-Rate check item is ignored.

Framed-Protocol

Framed-Protocol is primarily used as a reply item, but you can also use the Framed-Protocol check item in the user profile for PPP autodetection by the PortMaster:

bob  
Auth-Type = System, Framed-Protocol = PPP

See “Configuring a PPP or SLIP User” on page 4-25 for more information.

Group

You can define a group of users to restrict authentication when specifying Auth-Type = System. If a user profile contains the Group check item, only users that are defined as members of the specified group are authenticated.

The Group string consists of up to 63 printable, non-space, ASCII characters. The group must be contained within double quotation marks.

If you specify multiple groups in a user profile, the user must be a member of every group specified to be authenticated. In the following example, user ann is authenticated only if ann is a member of both the engineer group and the hardware group:

ann  
Auth-Type = System, Group = "engineer", Group = "hardware"

On UNIX hosts, groups are defined in /etc/group or via NIS. Refer to your system documentation for instructions on creating groups and adding members to groups.
Service-Type

Service-Type is generally used as a reply item, but it can be used with either of the following values as a check item. See “Specifying the Type of Service” on page 4-22 for more information on the Service-Type reply item.

Call-Check

You can authenticate a user before the NAS accepts the call by specifying the Call-Check value for Service-Type.

5105551234 Service-Type = Call-Check, Calling-Station-Id = "5105551234"

You can determine whether to answer a call based upon the values for the Calling-Station-Id or the Called-Station-Id check items. Some example uses of Call-Check are the following:

- Service class support
  
  For example, a service provider can offer toll-free dial-up service and minimize toll charges incurred over the toll-free number. Call-Check allows the service provider to answer the call based upon the Calling-Station-Id (the number from which the call originated). If a call does not come from a calling station assigned to the toll-free service, the PortMaster returns a busy signal rather than accepting the call. The service provider avoids incurring charges associated with answering the call if the caller (calling station) is not defined for this service. This service also provides an additional means of security by ensuring that a user dials in from a specified number, removing the need for dialback users.

  – Another example of a service class support is guest dial-up, where a service provider allows guests to dial up via a specified called number—Called-Station-Id.

- TCP-Clear tunneling

  TCP-Clear tunnels are similar to Layer 2 Tunneling Protocol (L2TP) tunnels, but are available only on PortMaster products. TCP-clear tunnels can be established for SLIP, asynchronous, and PPP connections.

  You can enable the call-check feature on a PortMaster supporting an ISDN Primary Rate Interface (PRI), such as the PortMaster 3 or PortMaster 4. The setup packet sent by the telephone company or line provisioner to the NAS includes the Called-Station-Id (the number dialed), Calling-Station-Id (the number the user is dialing from), or both. These
numbers are available over the D channel of a PRI. The PortMaster sends a RADIUS access-request to the server for any incoming call before accepting the call. The access-request packet includes the following:

- Attribute User-Name set to the value of the Calling-Station-Id, if known, or to “No-Call-Id”.
- Attribute Service-Type set to Call-Check
- Attribute Called-Station-Id set to the number called

For example, a PortMaster 3 supports call-check and expects to receive one of the following replies to an access-request:

- RADIUS access-accept with the reply items to configure the session, such as connecting the user via a netdata connection to a given host and TCP port.
- RADIUS access-accept with no reply items. This reply instructs the NAS to accept the call and apply the usual RADIUS authentication process to the login.
- RADIUS access-reject to reject the call.

**Framed-User**

You can use the Service-Type = Framed-User check item to authenticate users who make their connections using PPP.

**Outbound-User**

Use the Service-Type = Outbound-User check item to restrict a user to outbound accesses to network device ports. You must use the same attribute and value—Service-Type = Outbound-User—as a reply item in the user profile. See “Granting a User Outbound Telnet Access” on page 4-34 for more information.
Reply Items

Reply items can authorize or apply any of the following: type of service provided, callback information, routing information, connection protocol, timeouts, port limits, menus, maximum MTU, filters, remote login information, and termination menus. Table 4-2 summarizes the reply items you can include in user profiles.

### Table 4-2  User Profile Reply Items

<table>
<thead>
<tr>
<th>Item</th>
<th>Options</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callback-Id</td>
<td>Location name in double quotation marks (&quot; &quot;)</td>
<td>Specify only for Service-Type = Callback-Framed-User. Location must be in PortMaster location table.</td>
</tr>
<tr>
<td>Callback-Number</td>
<td>Phone number in double quotation marks (&quot; &quot;)</td>
<td>Specify only for Service-Type = Callback-Login-User.</td>
</tr>
<tr>
<td>Filter-Id</td>
<td>Filter name</td>
<td>Filter name to be used for packet or access filtering on the interface.</td>
</tr>
<tr>
<td>Framed-Compression</td>
<td>None</td>
<td>If this reply item is omitted, Van Jacobson TCP/IP header compression is used.</td>
</tr>
<tr>
<td>Framed-IP-Address</td>
<td>IP Address</td>
<td>The user’s IP address.</td>
</tr>
<tr>
<td>Framed-IP-Netmask</td>
<td>Netmask</td>
<td>The user’s netmask.</td>
</tr>
<tr>
<td>Framed-IPX-Network</td>
<td>Dotted decimal IPX network number</td>
<td>IPX network number.</td>
</tr>
<tr>
<td>Framed-MTU</td>
<td>Number</td>
<td>Number of bytes in maximum transmission unit (MTU).</td>
</tr>
<tr>
<td>Framed-Protocol</td>
<td>PPP</td>
<td>PPP is used for the connection. Can also be used as a check item.</td>
</tr>
<tr>
<td></td>
<td>SLIP</td>
<td>SLIP is used for the connection.</td>
</tr>
<tr>
<td>Item</td>
<td>Options</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Framed-Route</td>
<td>Destination IP address</td>
<td>The IP address of the destination network.</td>
</tr>
<tr>
<td></td>
<td>Gateway IP address</td>
<td>The IP address of the gateway to the destination network.</td>
</tr>
<tr>
<td></td>
<td>Metric</td>
<td>The number of routing hops to the destination network. Also known as the hop count.</td>
</tr>
<tr>
<td>Framed-Routing</td>
<td>None</td>
<td>Disables RIP on the interface.</td>
</tr>
<tr>
<td></td>
<td>Broadcast</td>
<td>The interface sends RIP updates.</td>
</tr>
<tr>
<td></td>
<td>Listen</td>
<td>The interface listens for RIP updates.</td>
</tr>
<tr>
<td></td>
<td>Broadcast-Listen</td>
<td>The interface sends and listens for RIP updates.</td>
</tr>
<tr>
<td>Idle-Timeout</td>
<td>In seconds</td>
<td>Specifies the idle time limit for a session.</td>
</tr>
<tr>
<td>Login-IP-Host</td>
<td>IP address</td>
<td>Address of the remote host.</td>
</tr>
<tr>
<td>Login-Service</td>
<td>Telnet</td>
<td>Establishes a Telnet connection to the remote host.</td>
</tr>
<tr>
<td></td>
<td>Rlogin</td>
<td>Establishes an rlogin connection to the remote host.</td>
</tr>
<tr>
<td></td>
<td>TCP-Clear</td>
<td>Establishes a TCP clear connection to the remote host.</td>
</tr>
<tr>
<td></td>
<td>PortMaster</td>
<td>Establishes a connection to the remote host using the PortMaster login service.</td>
</tr>
<tr>
<td>Login-TCP-Port</td>
<td>TCP port number</td>
<td>TCP port number of the Login-Service.</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu name in double quotation marks (&quot; &quot;)</td>
<td>Defines a menu in a user record. See Chapter 5, “Configuring RADIUS Menus.”</td>
</tr>
</tbody>
</table>
### Table 4-2  User Profile Reply Items (Continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Options</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-Limit</td>
<td>Number of B channels for ISDN Multilink PPP or Multilink V.120</td>
<td>Specifies the maximum number of B channels a user can use.</td>
</tr>
<tr>
<td>Session-Timeout</td>
<td>In seconds</td>
<td>Specifies the time limit for a session.</td>
</tr>
<tr>
<td>Service-Type</td>
<td>Administrative-User</td>
<td>Grants user full access to all configuration commands.</td>
</tr>
<tr>
<td>Callback-Login-User</td>
<td></td>
<td>Calls user back and connects via Telnet, <code>rlogin</code>, PortMaster, or TCP-Clear login service.</td>
</tr>
<tr>
<td>Callback-Framed-User</td>
<td></td>
<td>Calls user back and establishes a framed connection (PPP or SLIP). Location must be specified in PortMaster location table.</td>
</tr>
<tr>
<td>Framed-User</td>
<td></td>
<td>User uses PPP or SLIP for the connection. Can also be used as a check item.</td>
</tr>
<tr>
<td>Login-User</td>
<td></td>
<td>User connects via Telnet, <code>rlogin</code>, PortMaster, or TCP-Clear login service.</td>
</tr>
<tr>
<td>NAS-Prompt-User</td>
<td></td>
<td>Grants user limited access to commands (nonconfiguration only).</td>
</tr>
<tr>
<td>Outbound-User</td>
<td></td>
<td>User makes outbound connections via Telnet. Can also be used as a check item.</td>
</tr>
<tr>
<td>Termination-Menu</td>
<td>Menu name in double quotation marks (&quot; &quot;)</td>
<td>Menu to display after service is terminated.</td>
</tr>
</tbody>
</table>
Specifying the Type of Service

You must specify characteristics of the service that is provided to the user by specifying the desired Service-Type in each user profile. The reply items in each user profile determine how the user’s session is configured on the PortMaster. Table 4-3 defines each Service-Type value. Refer to the sections following the table for information on how and why you would use each one.

Table 4-3  Service-Type

<table>
<thead>
<tr>
<th>Service-Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative-User</td>
<td>The PortMaster grants the user a full administrative login—as if the user had logged in using <code>!root</code>. The user has full configuration ability and access to all operating system commands. This Service-Type is available only with ComOS 3.5 or later versions. For more information about this value, see “Granting a User Administrative Rights” on page 4-24.</td>
</tr>
<tr>
<td>Callback-Framed-User</td>
<td>The PortMaster verifies the user’s identity by disconnecting the port and dialing the user back using a specified location table entry. When the user’s identity is verified, PPP or SLIP is used for the connection. To specify the callback location, see “Using Callback to Authenticate a User” on page 4-24.</td>
</tr>
<tr>
<td>Callback-Login-User</td>
<td>The PortMaster verifies the user’s identity by disconnecting the port and dialing the user back at a specified number. The user’s identity must be verified before the connection is permitted. To specify the callback location, see “Using Callback to Authenticate a User” on page 4-24.</td>
</tr>
</tbody>
</table>
Reply Items

Table 4-3  Service-Type (Continued)

<table>
<thead>
<tr>
<th>Service-Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-Check</td>
<td>You can enable services without authenticating the user at the point of entry on a PortMaster that supports an ISDN Primary Rate Interface (PRI), such as the PortMaster 3. You must also configure the call-check feature on the PortMaster. For more information on this value, see “Call-Check” on page 4-17.</td>
</tr>
<tr>
<td>Framed-User</td>
<td>The user makes a connection via PPP or SLIP. See “Configuring a PPP or SLIP User” on page 4-25.</td>
</tr>
<tr>
<td>Login-User</td>
<td>The user connects via the Telnet, rlogin, or PortMaster service (in.pmd), or via TCP-Clear (netdata). See “Configuring a Login User” on page 4-31</td>
</tr>
<tr>
<td>NAS-Prompt-User</td>
<td>The PortMaster grants the user a limited administrative login. A PortMaster user can use the following commands: ifconfig, ping, ptrace, reboot, reset, set console, set debug, show, traceroute, and any nonconfiguration commands. The following commands are not permitted on a PortMaster: add, delete, erase, save, tftp, and any set commands except set console or set debug. This Service-Type is available only with ComOS 3.5 or later versions. See “Granting a User Administrative Rights” on page 4-24.</td>
</tr>
<tr>
<td>Outbound-User</td>
<td>The user uses Telnet for outbound connections. See “Granting a User Outbound Telnet Access” on page 4-34.</td>
</tr>
</tbody>
</table>

Note – If the RADIUS for UNIX server is used with a NAS other than a PortMaster product, the Administrative-User and NAS-Prompt-User Service-Types must not be used unless the other vendor’s implementation of these types is compatible with the Lucent
Remote Access implementation. The remainder of this appendix explains how to set reply items on a PortMaster NAS. Remember, you must own at least one PortMaster product to legally use Lucent RADIUS.

Granting a User Administrative Rights

You can grant a user administrative access to a PortMaster running ComOS 3.5 or later by specifying either of the following values for Service-Type:

- Administrative-User
- NAS-Prompt-User

**Administrative-User.** You can grant a user full PortMaster administrative ability by specifying Service-Type = Administrative-User. The user can configure the PortMaster client and can use all PortMaster commands.

```
bob                      Password = "ge55gep"
Service-Type = Administrative-User
```

**NAS-Prompt-User.** You can grant a user limited PortMaster administrative ability by specifying Service-Type = NAS-Prompt-User value. The user can use the following commands: `ifconfig`, `ping`, `ptrace`, `reboot`, `reset`, `set console`, `set debug`, `show`, and `traceroute`.

```
bob                      Password = "ge55gep"
Service-Type = NAS-Prompt-User
```

The user cannot configure the PortMaster client and cannot use these commands: `add`, `delete`, `erase`, `save`, `tftp`, and any `set` commands except `set console` or `set debug`.

Using Callback to Authenticate a User

You can authenticate a user with callback by specifying either of the following values for Service-Type:

- Callback-Framed-User
- Callback-Login-User
Callback-Framed-User. When a user’s Service-Type is Callback-Framed-User, you must specify a location using the Callback-Id reply item.

elfego  Password = ”ke$&54su"
    Service-Type = Callback-Framed-User,
    Callback-Id = ”elfego_home"

After the RADIUS server authenticates elfego, it sends an access-accept message including the Callback-Id to the PortMaster. The PortMaster checks its local location table; if there is a matching location name, it makes the connection using that location’s settings.

Note – To create location table entries, see the information on configuring dial-out locations in the PortMaster Configuration Guide.

Callback-Login-User. When a user’s Service-Type is Callback-Login-User, you must specify a telephone number using the Callback-Number reply item.

elfego  Password = ”ke$&54su"
    Service-Type = Callback-Login-User,
    Callback-Number = ”9,1-800-555-1234"

After the RADIUS server authenticates the user, it sends an access-accept message including the Callback-Number to the PortMaster. The PortMaster calls the user back at the specified number. The PortMaster follows the ATDT command set. It ignores hyphens in the number received from RADIUS and treats commas as pauses. If the user is reached successfully, the PortMaster prompts the user to reenter the password and then sets up the connection.

When a user’s Service-Type is Callback-Login-User or Login-User, you can supply additional information:

- Service used to connect to the host—See “Configuring a Login User” on page 4-31.
- Name or IP address of the remote host—See “Login-IP-Host” on page 4-32.
- TCP port number—See “Login-TCP-Port” on page 4-33.

Configuring a PPP or SLIP User

Specify Service-Type = Framed-User if the user is making the connection via PPP or SLIP. You must add the Framed-Protocol reply item to the user profile and specify whether PPP or SLIP is used.
Framed-Protocol can also be used as a check item for PPP autodetection by the PortMaster. See “Framed-Protocol” on page 4-16 for more information on the Framed-Protocol check item.

You can specify a packet filter to be used for each PPP or SLIP session, as shown here. See “Applying Packet Filters” on page 4-30 for more information. Access filters are applied to login users; see “Applying Access Filters” on page 4-34 for information.

Use the Framed-MTU reply item to configure the number of bytes in the maximum transmission unit (MTU) for a user’s connection. Framed-MTU is used only for PPP and SLIP connections. For PPP connections, Framed-MTU can be between 100 and 1520 bytes. SLIP connections can have an MTU between 100 and 1006 bytes. On IPX networks, set Framed-MTU to at least 600 bytes.

**Note** – If PPP negotiates an MTU for the connection, the Framed-MTU setting is ignored.

### Enabling and Disabling Compression

Van Jacobson TCP/IP header compression is enabled by default. To disable compression, set the Framed-Compression value to **None**.

```
Framed-Compression = None
```

To reenable compression, set the Framed-Compression value to **Van-Jacobson-TCP-IP**.

```
Framed-Compression = Van-Jacobson-TCP-IP
```
Specifying an IP Address for the User

Use the Framed-IP-Address reply item to specify the user's IP address:

```
jake  Auth-Type = System
      Service-Type = Framed-User,
      Framed-Routing = None,
      Framed-Protocol = PPP,
      Framed-IP-Address = 172.28.1.1
```

When Framed-IP-Address is set to 255.255.255.255, the PortMaster negotiates the address with the end node (dial-in user). When it is set to 255.255.255.254 (or omitted), the PortMaster assigns an IP address to the dial-in user from the assigned address pool.

**Note** – To create an assigned address pool for the PortMaster, use the `set assigned_address Ipaddress` command on the PortMaster, where `Ipaddress` is the first IP address in the address pool. See the PortMaster Configuration Guide for more information on assigned address pools.

Applying a Subnet Mask to the Address

Use the Framed-IP-Netmask reply item as follows to specify a subnet mask. The subnet mask is applied to the address specified for the user in the Framed-IP-Address reply item. The PortMaster uses the specified value to update its routing table when the user logs in.

```
jake  Auth-Type = System
      Service-Type = Framed-User,
      Framed-Routing = None,
      Framed-Protocol = PPP,
      Framed-IP-Address = 192.168.10.232,
      Framed-IP-Netmask = 255.255.255.192
```

In this example, a netmask of 255.255.255.192 is used. The user is allocated a 64-host subnet. The subnet allocated to `jake` includes hosts with addresses from 192.168.10.193 through 192.168.10.255.
If this reply item is omitted, the default subnet mask of 255.255.255.255 is used. Use the Framed-IP-Netmask reply item with caution because it affects both routing and Proxy Address Resolution Protocol (Proxy ARP) on the PortMaster. Using Framed-IP-Netmask adds a temporary route to the routing table on the PortMaster. See the PortMaster Routing Guide for more information.

Note – This reply item requires ComOS 3.5 or later. You must use the set user-netmask on command to enable the PortMaster to use the netmask value. If the command set user-netmask off has been issued on the PortMaster, the default subnet mask of 255.255.255.255 is applied to all connections regardless of what the RADIUS server returns. Before using this reply item, read about the set user-netmask command in the PortMaster Configuration Guide or the PortMaster Command Line Reference.

Adding a Route to the PortMaster Routing Table

Use the Framed-Route reply item to add a route to the PortMaster routing table when service to the user begins. Three pieces of information are required: the destination IP address, gateway IP address, and metric (hop count).

```
jake
Auth-Type = System
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Framed-IP-Address = 172.28.1.1,
Framed-Route = "172.28.1.0 172.28.1.1 1"
```

In this example, 172.28.1.0 is the IP address of a destination network, 172.28.1.1 is the IP address of the gateway for this network, and 1 is the metric.

If 0.0.0.0 is specified as the gateway IP address, the user’s specific IP address is substituted for the gateway.

In ComOS 3.5 or later, you can use the classless interdomain routing (CIDR) format for the Framed-Route, which identifies the number of high-order bits in the destination IP address, as shown in the following example.

```
jake
Auth-Type = System
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Framed-IP-Address = 172.28.1.1,
Framed-Route = "172.28.1.0/28 172.28.1.1 1"
```
Configuring RIP on the User’s Interface

Use the Framed-Routing reply item to control how Routing Information Protocol (RIP) is used on the user’s interface. Table 4-4 explains RIP options.

Table 4-4 Framed-Routing Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Disables RIP on the interface.</td>
</tr>
<tr>
<td>Broadcast</td>
<td>The interface sends RIP updates.</td>
</tr>
<tr>
<td>Listen</td>
<td>The interface listens for RIP updates.</td>
</tr>
<tr>
<td>Broadcast-Listen</td>
<td>The interface sends and listens for RIP updates.</td>
</tr>
</tbody>
</table>

In the following example, Framed-Routing is set to None so that the interface neither sends nor listens for RIP updates.

```
sri Password = "4r2tkgbp"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP
```

The usefulness of the Idle-Timeout reply item is reduced if RIP is active on the user’s interface because updates are sent every 30 seconds, keeping the port active rather than idle. See “Idle-Timeout” on page 4-36 for information on the Idle-Timeout reply item.

Typically, Framed-Routing is set to None for user connections, and is set to Broadcast or Broadcast-Listen for connections to routers that require routing updates via RIP. See the PortMaster Routing Guide and the PortMaster Command Line Reference for more information.

Configuring an IPX Network Connection

When an IPX network is used for a particular user’s connection, you must include the Framed-IPX-Network reply item in the user profile. The PortMaster supports IPX over PPP.

See the PortMaster Configuration Guide for more information on CIDR.
Specify Framed-IPX-Network in dotted decimal notation (xx.xx.xx.xx). For example, the hexadecimal network number 123456 must be expressed as 0.18.52.86.

ajit     Password = "testing"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Framed-IPX-Network = 0.18.52.86

On a UNIX system, the following Perl script converts an IPX hexadecimal network number to dotted decimal notation:

```perl
#!/usr/local/bin/perl
# hex   - convert ip addresses to hexadecimal and vice versa
for (@ARGV) {
    if (/\./) {             # convert . to hex
        @octets = split(\./,$_);
        for $octet (@octets) {
            printf "%02X",$octet;
        }
        print "\n";
    } else {                # convert hex to .
        $buf = '';           
        while (s/\w\w//) {
            $buf .= hex($&).'.';
        }
        $buf =~ s/.$/\n/;  
        print $buf;
    }
}
```

**Applying Packet Filters**

Use the Filter-Id reply item to associate packet filters with each PPP or SLIP user authenticated with RADIUS. In the following example, the firewall filter is used during a connection:

rakshah    Password = "yj8hg355"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Filter-Id = "firewall"
For the Filter-Id attribute to initiate filtering, the filter must be previously defined on the PortMaster and the rule set must contain at least one rule. If you specify a Filter-Id in the user profile, but do not define the filter, then no filtering is performed. If you define the filter, but do not create any rules in the filter, then no filtering is performed.

You must define filters in the filter table on each PortMaster that the user accesses, unless you are using ChoiceNet. See the ChoiceNet Administrator’s Guide for information on ChoiceNet and how it provides storage for filters in a central site.

To control whether the filter restricts incoming or outgoing traffic, the filter defined on the PortMaster must have an .in or .out suffix attached to its name. In the previous example, the filter firewall.in is used as a filter for packets entering the PortMaster via the interface, and firewall.out is used as an output filter for packets leaving the PortMaster via the interface.

Do not specify the .in and .out suffixes in the user profile. When a user dials in to the PortMaster, the .in or .out suffix is automatically appended to the filter name provided by RADIUS for UNIX.

Note – To configure filters on a PortMaster, see the information on configuring filters in the PortMaster Configuration Guide. Filters specified in RADIUS for UNIX can also be dynamically loaded via ChoiceNet. For more information see the ChoiceNet Administrator’s Guide.

Configuring a Login User

If the user is logging in to your system, specify either of the following values for Service-Type:

- **Callback-Login-User**
  
  Specify Callback-Login-User when you want to call the user back to at a specified telephone number before authorizing service. See “Using Callback to Authenticate a User” on page 4-24.

- **Login-User**
  
  Specify Login-User for all other login connections:

  ```
  sri
  Auth-Type = System
  Service-Type = Login-User
  ```

  You can specify an access filter to be used for each login session. See “Applying Access Filters” on page 4-34.
When a user’s Service-Type is Callback-Login-User or Login-User, you can supply additional information:

- Service used to connect to the host
- Name or IP address of the remote host
- TCP port number.

**Login-Service**

When a user’s Service-Type is Login-User or Callback-Login-User, you can use the Login-Service reply item to specify the service used to connect to the host.

If you do not use this reply item, the PortMaster login service is used by default. Table 4-5 describes all Login-Service values.

<table>
<thead>
<tr>
<th>Login-Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telnet</td>
<td>Establishes a Telnet connection to the remote host. Port 23 is the default.</td>
</tr>
<tr>
<td>Rlogin</td>
<td>Establishes an rlogin connection to the remote host.</td>
</tr>
<tr>
<td>TCP-Clear</td>
<td>Establishes a TCP clear connection to the remote host. 8-bit data is passed through this connection without interpretation. This option is the equivalent of the netdata login service on the PortMaster. Port 6000 is the default.</td>
</tr>
<tr>
<td>PortMaster</td>
<td>Establishes a connection to the remote host using the PortMaster login service. To use this setting with UNIX versions of RADIUS, you must install the in.pmd daemon on the remote host.</td>
</tr>
</tbody>
</table>

**Login-IP-Host**

When a user’s Service-Type is Login-User or Callback-Login-User, you can use the Login-IP-Host reply item to specify the name or IP address of the remote host. You can specify more than one host by using multiple Login-IP-host attributes.
If you do not use this reply item to specify a remote host, the PortMaster default host is used. See the PortMaster Command Line Reference for information on setting the default host.

In following example, the user is authenticated and then called back at the Callback-Number. If this call is successfully authenticated, a Telnet connection to host 192.168.1.76 is established.

    uma Password = "gk4u4p"
    Service-Type = Callback-Login-User,
    Login-IP-Host = 192.168.1.76,
    Login-Service = Telnet,
    Callback-Number = "9,1-800-555-1234"

If Login-IP-Host is set to 0.0.0.0 or omitted, the host defined for the port in the PortMaster is used. If Login-IP-Host is set to 255.255.255.255, the user must enter the hostname or the host’s IP address at the host: prompt that appears.

If the user is to log in to a particular TCP port on the remote host, specify the port number with the Login-TCP-Port reply item. See “Login-TCP-Port” on page 4-33.

**Login-TCP-Port**

When a user’s Service-Type is Login-User or Callback-Login-User, you can use the Login-TCP-Port reply item to specify the port number if the user is to log in to a particular TCP port on the remote host. This reply item is often used with the Outbound-User reply item (see “Granting a User Outbound Telnet Access” on page 4-34) and the Login-IP-Host reply item (see “Login-IP-Host” on page 4-32).

In following example, the user is authenticated and then called back at the Callback-Number. If authentication is successful, a Telnet connection to port 6220 on host 192.168.1.76 is established.

    uma Password = "gk4u4p"
    Service-Type = Callback-Login-User,
    Login-IP-Host = 192.168.1.76,
    Login-Service = Telnet,
    Login-TCP-Port = 6220,
    Callback-Number = "9,1-800-555-1234"

If Login-TCP-Port is omitted, the port defined for Telnet service on the PortMaster is used. The default port number for Telnet is 23.
Applying Access Filters

An access filter is a filter associated with a login user. Use the Filter-Id reply item to associate an access filter with a host prompt login user authenticated with RADIUS for UNIX. In the following example, the `restricthost` filter is used to restrict the hosts that the user can access during a connection:

```
sunil     Password = "76patel15rj"
Service-Type = Login-User,
Login-IP-Host = 255.255.255.255,
Login-Service = Telnet,
Login-TCP-Port = 23,
Filter-Id = "restricthost"
```

**Note** – Access filters only restrict the host(s) a user can access. They do not restrict what the user can access from that host.

For the Filter-Id attribute to initiate filtering, the filter must be previously defined and the rule set must contain at least one rule. If you specify a Filter-Id in the user profile, but do not define the filter, then no filtering is performed. If you define the filter, but do not create any rules in the filter, then no filtering is performed.

You must define access filters in the filter table on each PortMaster the user accesses, using the same name as the Filter-Id. The access filter name defined in the user profile must be exactly the same as the filter name defined on the PortMaster. The PortMaster does not append an extension to the name of an access filter, unlike packet filters.

**Note** – ChoiceNet cannot be used with access filters.

Granting a User Outbound Telnet Access

Specify Service-Type = Outbound-User to enable a user to gain outbound access to network device ports using Telnet. This feature is supported in ComOS version 3.3.2 or later. To use this feature, you must set the relevant asynchronous ports on your PortMaster as either host-controlled devices, using the command `set S0 device Device network Mode`, or as devices capable of two-way operation, using the command `set S0 twoway Device network Mode`. See the PortMaster Command Line Reference for more information on setting the PortMaster port type.
To restrict users to outbound access, the user profile must include this same attribute and value—Service-Type = Outbound-User—as a check item. The Login-TCP-Port setting can be used to specify the TCP port for the connection; the port number must be between 10000 and 10100 inclusive.

```
roxy Password = "ge55gep", Service-Type = Outbound-User
Service-Type = Outbound-User,
Login-Service = Telnet,
Login-TCP-Port = 10000
```

In this example, when `roxy` attempts an outbound connection, the PortMaster client checks its local user table for an entry for the user. If `roxy` is not found in the user table, the PortMaster sends an access-request to the RADIUS server indicating that `roxy` is an Outbound-User.

The RADIUS for UNIX server examines `roxy`'s profile in the **users** file. If Outbound-User is included as a reply item, the PortMaster is notified to permit the connection.

Configure the PortMaster as shown in the following example. This example configures port S1; however, you can configure multiple ports to listen at different TCP port numbers or at the same TCP port number to create a pool of devices.

```
Command> set s1 device /dev/network
Command> set s1 service_device telnet 10000
Command> set s1 modem off
```

### Setting Timeouts

You can apply the following two kinds of timeouts to any session:

- **Idle-Timeout**
- **Session-Timeout**

**Note** – Idle-Timeout and Session-Timeout values are specified in **seconds** in the RADIUS user profiles. Timeout values set directly on the PortMaster using the PortMaster command line interface or the PMVision™ graphical user interface (GUI) are specified in **minutes**, by default.
Idle-Timeout

Use Idle-Timeout to specify the number of seconds a session can be idle before it is disconnected. Idle-Timeout can range from 2 seconds to 14400 seconds (4 hours) and is rounded down to a multiple of 60 if greater than 240.

**Note** – An Idle-Timeout value of 0 in the user profile is overridden by the Idle-Timeout setting configured on the PortMaster. Any Idle-Timeout value in the user profile greater than zero overrides the Idle-Timeout setting configured on the PortMaster.

In this example, if the session is inactive longer than 600 seconds (10 minutes), *greta* is disconnected.

```
greta    Password = "ge55gep"
            Service-Type = Framed-User,
            Framed-Routing = None,
            Framed-Protocol = PPP,
            Idle-Timeout = 600
```

**Note** – The effectiveness of the Idle-Timeout reply item can be reduced if RIP is active on the user’s interface because updates are sent every 30 seconds, keeping the port active rather than idle. See “Configuring RIP on the User’s Interface” on page 4-29 for information on using the RIP reply item.

Session-Timeout

Use Session-Timeout to specify the time limit for a session. If this reply item appears in a user profile, the user is disconnected when the time limit is reached. Session-Timeout is specified as a particular number of seconds, up to a maximum of 31536000 (1 year).

In the following example, *adil* is automatically disconnected after 7200 seconds (2 hours):

```
adil    Password = "khan235f3"
            Service-Type = Framed-User,
            Framed-Routing = None,
            Framed-Protocol = PPP,
            Session-Timeout = 7200
```

Suppose you want to impose a time limit on connections to a particular port. For example, you want to provide a means for users to only check their email (using a filter) and don’t want the users tying up the port. In the following example, users connected
on port 10 of NAS 10.10.10.1 are automatically disconnected after 600 seconds (10 minutes):

```
DEFAULT Auth-Type = System, NAS-IP-Address = 10.10.10.1, NAS-Port = 10
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Session-Timeout = 600,
Filter-Id = "emailonly"
```

**Using Menus**

You can specify menus to be used when the user is authenticated or when the user ends a session. See Chapter 5, “Configuring RADIUS Menus,” for more information.

**Menu**

Use the Menu reply item to call a menu by reference. The Menu reply item is the only reply item in the user profile when a menu is referenced.

```
DEFAULT Auth-Type = System
Menu = "menu1"
```

In this example, after the user is authenticated, the `menu1` menu is displayed and the user is prompted to make a selection. When the user selects a menu option, the corresponding service is provided.

**Termination-Menu**

Use Termination-Menu to present a menu to the user when the service ends. If a Termination-Menu reply item is not included in the user profile, the user is disconnected immediately after a SLIP, PPP, or login session.

```
mia Password = "soon86yee"
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Termination-Menu = "mainmenu"
```

**Note** – If you want to disconnect the line when the service ends, do not use the Termination-Menu attribute.
Controlling the Number of Available Ports

Use the Port-Limit reply item to control the maximum number of ports available for a Multilink PPP or Multilink V.120 connection. Port-Limit applies only to ISDN and (with ComOS 3.8 or later) asynchronous connections; other connection types are not affected.

The Port-Limit value can be as high as the maximum number of B channels available for the ISDN ports. For example, if a PortMaster has 15 ISDN BRI ports, the Port-Limit value can be as high as 30.

In the following example, red’s connection can use only one B channel:

red Password = "9pg$lac", NAS-Port-Type = ISDN
Service-Type = Framed-User,
Framed-Routing = None,
Framed-Protocol = PPP,
Port-Limit = 1

Note – The Port-Limit reply item only limits multilink connections. It does not prevent simultaneous logins.

Using RADIUS with PAP and CHAP

You can use RADIUS with Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP).

PAP

The PortMaster sends the PAP ID and password to the RADIUS server in an access-request packet as the User-Name and User-Password. The PortMaster includes the Service-Type = Framed-User and Framed-Protocol = PPP attributes in the request as a hint to the RADIUS server that PPP service is expected.

To authenticate a user with PAP, user profiles can include Auth-Type = Local, Auth-Type = System, or Auth-Type = SecurID.
CHAP

For CHAP, the PortMaster generates a random challenge and sends it to the user. The user returns a CHAP response, CHAP ID, and CHAP username. The PortMaster then sends an access-request packet to the RADIUS server with the CHAP username as the User-Name and with the CHAP ID and CHAP response as the CHAP-Password. The random challenge can either be included in the CHAP-Challenge attribute or, if it is 16 octets long, it can be placed in the Request Authenticator field of the access-request packet. The PortMaster includes the attributes Service-Type = Framed-User and Framed-Protocol = PPP as a hint to the RADIUS server that PPP service is expected.

The RADIUS server does the following:

1. Looks up a password based on the User-Name.
2. Uses MD5 to encrypt the password, the CHAP ID octet, and the CHAP challenge.
3. Compares the result to the CHAP-Password.
4. If the items in Steps 2 and 3 match, the server sends an access-accept packet to the PortMaster. If there is no match, the server sends back an access-reject packet.

CHAP requires that the user’s password be available on the RADIUS host in unencrypted (clear text) format so that the server can encrypt the CHAP challenge and compare the result to the CHAP response. If the password is not available in clear text, the server sends an access-reject to the client.

To force all PPP communication to use CHAP authentication, do the following:

1. **Set the Auth-Type = Local.**
   
   This is the default value.

2. **Set passwords in clear text; they must not be encrypted.**

3. **Turn off PAP and turn on CHAP by using the following commands on the PortMaster:**
   
   Command> set pap off
   Command> set chap on
Configuring Database Caching of User Profiles

RADIUS offers database support for caching user profiles to increase the speed and efficiency of user lookups. Lucent Remote Access recommends caching user profiles when the users file contains more than 500 users.

The **builddbm** utility included with RADIUS for UNIX converts the users text file to the UNIX DBM format, which increases the speed of user lookups.

To run **builddbm**, use the following commands:

```
    cd /etc/raddb
    ./builddbm
```

RADIUS uses the new database when it receives its next authentication request.

To run the **radiusd** daemon after the users file is converted to DBM, run **radiusd** with the -b option:

```
    /etc/radiusd -b
```

The **builddbm** utility generates **users.dir** and **users.pag** files that are used by the **radiusd** daemon. On some versions of UNIX a single **users.db** file is created instead. If duplicate users file profiles are present, **builddbm** prints the number of profiles and identifies the line number of the duplicate profiles.

**Note** – After the users file has been converted to the .dbm format, you must run **builddbm** again if you make any changes to the user profiles.

Example PPP User Profile

User profiles can be configured in a number of ways to fit network security requirements. The following example illustrates a typical RADIUS profile for a PPP user:

```
masha
    Password = "ge55gep"
    Service-Type = Framed-User,
    Framed-Routing = None,
    Framed-Protocol = PPP,
    Framed-IP-Address = 255.255.255.254,
    Framed-Compression = Van-Jacobson-TCP-IP,
    Framed-MTU = 1500,
    Filter-Id = "firewall"
```
Example PPP User Profile

In this example, user masha has password ge55gеп. She is a Framed-User, which indicates that she uses SLIP or PPP for her connections. In the following line, Framed-Protocol is specified as PPP.

An IP address of 255.255.255.254 is specified, indicating that an IP address is assigned to masha from the PortMaster assigned address pool.

Note – To create an assigned address pool, see the PortMaster Configuration Guide.

Framed-Routing is set to None, which disables RIP for masha’s interface. RIP packets are not sent or listened for. Van Jacobson TCP/IP compression is used for the connection, and the MTU is set to 1500 bytes.

The Filter-Id identifies the packet filter(s) used for the connection if any are defined on the PortMaster or in ChoiceNet; firewall.in is used as an input filter and firewall.out is used as an output filter.
RADIUS menus enable a user to select different login options after being authenticated. The user only needs a single username for all options rather than a different username for each connection option.

RADIUS menus are implemented as text files located in the menus subdirectory on the RADIUS server, typically /etc/raddb/menus. The number of menu files under the menus directory is unlimited. A menu file can accommodate up to 2KB of display data plus menu selection entries. Menus can refer to other menus.

**Menu File Format**

A menu file consists of the following elements:

- An initial line containing only the keyword **menu**
- Additional lines of text to be displayed to the user
- A line containing only the keyword **end**
- One or more menu selection entries
- The DEFAULT menu selection entry

The **menu** and **end** keywords indicate the start and end of the text displayed to the user. Text between the **menu** and **end** keywords can be any printable ASCII characters up to a maximum of 2Kb. The text in the menu file is case-sensitive.

Each menu selection entry consists of the menu choice shown at the beginning of a line, followed by one or more lines of reply items—one per line—starting with spaces or tabs. You can enter comments among the menu selection entries by starting each comment line with a number sign (#).

The special menu choice **DEFAULT** must be the last menu selection entry. The DEFAULT menu is called when the user enters no choice or a choice that does not match a menu selection entry in the menu file.

Use the special menu choice **EXIT** for a menu selection—such as “Quit”—that disconnects the user.
**Single-Level Menu**

A single-level menu does not refer to other menus. The following example shows a file named `/etc/raddb/menus/menu_welcome` for a single-level menu with three options:

```
menu
  *** Welcome to EDU OnLine ***
  Please select an option:

  1.  Start SLIP session
  2.  Start PPP Session
  3.  Quit

  Option:
end
# This is a single-level menu called menu_welcome
1
  Service-Type = Framed-User,
  Framed-Protocol = SLIP,
  Framed-IP-Address = 255.255.255.254,
  Framed-Routing = None,
  Framed-MTU = 1006,
  Termination-Menu = "menu_welcome"
#
2
  Service-Type = Framed-User,
  Framed-Protocol = PPP,
  Framed-IP-Address = 255.255.255.254,
  Framed-Routing = None,
  Termination-Menu = "menu_welcome"
#
3
  Menu = "EXIT"
#
DEFAULT
  Menu = "menu_welcome"
```

In the single-level menu example, after RADIUS authenticates the user, `menu_welcome` is displayed and the user is prompted to select a service from this menu. Once the user has finished the SLIP or PPP session, the termination menu—also `menu_welcome` in this case—is displayed and the user is prompted to select a new
service. If a Termination-Menu reply item is not included in the list of reply items corresponding to the user's menu selection, the user is disconnected immediately after the SLIP or PPP session.

**Nested Menus**

Nested menus refer to other menus. In the following example menu file, the menu that the user sees has an `other` option; if selected, this option displays a second menu:

```plaintext
menu
*** Welcome to the Internet Service ***
Please enter an option:
  ppp - Start PPP session
  telnet - Begin login session with a host
  other - Display a second menu
Option:
end
# This is a nested menu called menu_internet
ppp
  Service-Type = Framed-User,
  Framed-Protocol = PPP,
  Framed-IP-Address = 255.255.255.254,
  Framed-Routing = None,
  Framed-MTU = 1500
#
telnet
  Service-Type = Login-User,
  Login-IP-Host = 172.16.1.81,
  Login-Service = Telnet,
  Login-TCP-Port = 23
#
other
  Menu = "menu_other"
#
DEFAULT
  Menu = "menu_internet"
```
Termination Menus

Termination menus are presented to users when their service ends. The termination menu value must be contained within double quotation marks. The Termination-Menu reply item in the user's profile calls the menu.

With a user profile as shown in the following example, user *bob* sees **menu1** when he finishes his PPP session. When *bob* selects a menu option, the corresponding service is provided. He can begin another session or he can quit.

```
bob
  Password = "ge55gep"
  Service-Type = Framed-User,
  Framed-Protocol = PPP
  Framed-IP-Address = 255.255.255.254,
  Framed-Routing = None,
  Termination-Menu = "menu1"
```

Menus Called by Reference

Any user profile in the **users** file—including the DEFAULT profile—can call a menu by reference. The Menu reply item is the only reply item in the user profile when a menu is referenced.

```
DEFAULT
  Auth-Type = System
  Menu = "menu1"
```

In this example, after any user is authenticated via the DEFAULT profile, the **menu1** menu is displayed and the user is prompted to make a selection. When the user selects a menu option, the corresponding service is provided.

Menu Filenames

You must create the menu filename in the **menus** subdirectory on the RADIUS server, described on page 5-1. Menu names can consist of up to 120 printable, non-space, ASCII characters; in the **users** file the menu name must be enclosed in double quotation marks (" ").
ActivCard, Inc. provides an additional level of security in user identification and
authentication by using ActivCard tokens to generate codes and ActivEngine software to
process the codes. This system of software and hardware authentication is often referred
to as ActivCard.

This chapter is an overview of the installation and basic configuration of ActivEngine
and ActivAdmin software when used with RADIUS. This chapter includes the following
topics:

- “Overview of ActivCard Components” on page 6-2
- “How ActivCard Works with RADIUS” on page 6-3
- “Installing the ActivEngine Components on a UNIX Host” on page 6-4
- “RADIUS Configuration for ActivCard” on page 6-7
- “Troubleshooting ActivCard” on page 6-9

This information is intended to serve as a quick reference guide for the ActivEngine and
ActivAdmin software. Refer to the ActivCard manual set for detailed features of
ActivCard and future ActivCard software releases.

Note – Lucent Remote Access Technical Support does not provide support for
ActivEngine and ActivAdmin installation and configuration. See Appendix E, “Contact
Information for Third-Party Products,” for information on how to contact ActivCard,
Inc. Lucent Remote Access Technical Support provides support for RADIUS when used
with ActivCard products only after you have verified that all ActivCard components are
working properly.

The ActivCard software version 2.2 has the following requirements:

- Solaris 2.4 or Solaris 2.5 platform
- Standard UNIX TCP/IP network protocol
- 5MB free disk space for the ActivEngine software
Recommended disk space is 10MB for the ActivEngine database—necessary size depends on the number of registered tokens and users.

Note – ActivEngine software is also supported on Windows NT platforms. RADIUS server 2.1 is currently available only on UNIX platforms.

Overview of ActivCard Components

The ActivCard authentication system (generally referred to simply as ActivCard) consists of the following components:

- **ActivEngine server**
  
  Processes the username (referred to as *login ID* in ActivCard documentation) and a token-generated dynamic password entered by the user. ActivEngine then authenticates the user and either provides or rejects access to the secured application.

- **ActivAdmin**
  
  Software that initializes, assigns, unlocks, and resynchronizes tokens; manages the ActivEngine database of users and tokens; and applies and revokes access privileges. ActivAdmin includes standard token profiles for Master tokens and for ActivCard server solutions. ActivAdmin functionality includes:

- **Token**
  
  A small, handheld device that generates dynamic (one-time-use) passwords for user authentication. ActivCard tokens can generate dynamic passwords in two modes:

  - Asynchronous or challenge/response mode. The token generates a dynamic password based on the challenge—issued by the ActivEngine server when the user logs in—entered on the token by the user. The user then enters the password on the UNIX system to complete authentication.

  - Synchronous mode. The token generates a dynamic password based on its authentication counter. When the user enters this password on the system, the ActivEngine server compares it with the password the server generates based on the authentication counter stored in its database for that token.

  ActivCard tokens have an optical interface that enables automated authentication. ActivCard supports several token models with different features, such as secret value extraction, server authentication, and data certification.
How ActivCard Works with RADIUS

Note – To use RADIUS for UNIX version 2.1 with ActivCard, you must run the `sradiusd` daemon rather than `radiusd`. If you are using both ActivCard and iPass, contact support@livingston.com.

Determine which token profile to apply to each user needing access to the system or application secured by ActivCard. Use ActivAdmin to do the following:

- Initialize user tokens with the token profile
- Assign the initialized tokens to users
- Store the token data and assignment information in the ActivEngine database

You can install the ActivAdmin and ActivEngine components on the same host or on different hosts as long as the ActivEngine is accessible via TCP/IP.

When ActivCard is used with RADIUS, a connection proceeds as described below. This example assumes you have established a challenge-response keyword, enabling the user to choose between asynchronous and synchronous authentication.

1. A remote user initiates a connection by dialing in to the PortMaster.
2. The PortMaster prompts the user for a username and password.
3. The user enters the username and password.
   - If the user chooses asynchronous authentication, the password is the challenge-response keyword.
– If the user chooses synchronous authentication, the password is a one-time password generated by the token. The user first enters a memorized personal identification number (PIN) on the token and presses the **SECRET** key on the token keypad. The token, which is synchronized with the ActivEngine, displays the one-time password.

4. The PortMaster forwards the username and password to the RADIUS server for authentication.

5. The RADIUS server examines the **users** file, scanning for the appropriate username. When the profile is located, it is examined to determine the user’s authentication type.

6. When the RADIUS server discovers that the authentication type is ActivCard, it forwards the username and password to the ActivEngine.

7. ActivEngine examines its database for the username and verifies the user’s identity based on the password expected for the token.

   If the user entered the challenge-response keyword, ActivCard responds with a challenge string. The user enters a memorized personal identification number (PIN) on the token, presses the **AUTH** key on the token keypad, and enters the challenge. The token generates and displays a one-time password that the user enters to respond to the challenge. ActivEngine evaluates this response to verify the user.

8. ActivEngine sends the result of the database lookup (identity verified or not verified) to the RADIUS server.

9. If the user’s identity has been verified by ActivEngine, the RADIUS server sends an access-accept message to the PortMaster with the session configuration information from the RADIUS user profile. If ActivEngine has rejected the user, the RADIUS server sends an access-reject message to the PortMaster.

---

**Installing the ActivEngine Components on a UNIX Host**

This section summarizes the steps you perform to install the ActivEngine components. See the *ActivEngine for UNIX Systems Installation and Administration Guide* for detailed instructions.
1. **Determine the following information before beginning installation:**
   - Directory where you want to install ActivEngine. The default directory is `/usr/aeg`.
   - ActivEngine identifier—a unique name of up to 20 characters. Valid characters are the lowercase and uppercase alphabets, numerals 0-9, and the underscore (_) character.
   - IP address or Domain Name System (DNS) hostname of the system where you want to install the ActivEngine server. If you do not enter the IP address during installation, the DNS hostname of the current system is used by default.
   - Port number for authentication client connections. The default port number is 8866.
   - Port number for administration client connections. The default port number is 8867.
   - Whether you want the ActivEngine server to start automatically during a system reboot. If you choose automatic start, configure the ActivEngine server for automatic startup mode. Otherwise, the system asks you to extract the secret value each time the system is rebooted. Automatic startup mode is less secure than starting the server after each system reboot.
     
     If you choose not to configure the automatic startup mode during the installation process, you can choose to configure this mode later at any time. Refer to the *ActivEngine for UNIX Systems Installation and Administration Guide* for more information.
   - Name of the serial port on the host system to which the ActivCoupler is connected. The default port is `/dev/ttya`.

2. **Connect an ActivCoupler to the system hosting the ActivEngine server.**
   You initialize the Master token with the ActivCoupler.

3. **Install ActivEngine.**
   See the *ActivEngine for UNIX Systems Installation and Administration Guide* for instructions.

4. **Initialize the ActivEngine database with a Master token and initialize at least one Master token backup.**
Using the ActivEngine Test Utility

See the ActivEngine for UNIX Systems Installation and Administration Guide for instructions.

5. **Backup the ActivEngine database.**

See the ActivEngine for UNIX Systems Installation and Administration Guide for instructions.

6. **Use ActivAdmin to initialize end-user tokens with the appropriate token profile.**

See the ActivAdmin for UNIX Systems User Guide for instructions. A slot on each token must be assigned to RADIUS.

7. **Back up the ActivEngine database.**

See the ActivEngine for UNIX Systems Installation and Administration Guide for instructions.

8. **Use ActivAdmin to create a user record for each user that is assigned a token.**

See the ActivAdmin for UNIX Systems User Guide for instructions.

**Note** – ActivEngine usernames are case-sensitive.

9. **Assign the initialized tokens to users.**

See the ActivAdmin for UNIX Systems User Guide for instructions.

10. **Distribute the initialized tokens to the authorized users.**

**Using the ActivEngine Test Utility**

You can use the ActivEngine test utility `aegtest` to do the following:

- Verify that the RADIUS for UNIX client application is correctly configured in the ActivEngine database and can connect to the ActivEngine server.

- Test asynchronous and synchronous authentication for a token with a specific application.

Refer to the ActivCard, Inc. application note, *ActivEngine 2.x for UNIX Systems Test Utility* for instructions on using the utility.
RADIUS Configuration for ActivCard

The RADIUS server requires minimal configuration after you have installed the ActivCard software.

1. From the ActivCard software distribution, copy the file config.aeg.example to the /etc/raddb directory on the RADIUS server host.

2. Rename the file config.aeg.
   
   RADIUS uses the parameters described in /etc/raddb/config.aeg to connect to the ActivCard server.


4. Specify Auth-Type = ActivCard as a check item for all users to be authenticated with ActivCard.

5. Run sradiusd.
   
   If you are using ActivCard authentication, you cannot run radiusd.

Example config.aeg File

The text of an example config.aeg file is presented below. The comments explain the parameters that you specify in the file. Substitute values relevant to your installation for the values appearing in the parameters.

```plaintext
# Rename this file to config.aeg after installing new ActivCard server
# -----------------------------------------------------------------
# This file contains the configuration information necessary for the
# RADIUS server to connect to the ActivEngine, which is the
# ActivCard Authentication Server.
# -----------------------------------------------------------------
#
# -----------------------------------------------------------------
# ACTIVCARD_APPLICATION: APPLICATION_ID
#
# ActivCards contain up to four slots. Each slot contains a set
# of independent DES keys and parameters, so that in practice an
# ActivCard is equivalent to four «Tokens». Each of those «Tokens»
# is called a slot and can be used to authenticate through an application
```

Installing and Configuring ActivCard
# to a distinct ActivEngine. The concept that the ActivEngine uses to
decline which slot is to be used to verify a dynamic password (ultimately
which key among the set of keys stored for this token) is
the «Application»: Application (Server) -> Slot (Token)
#
# The following specifies the application to be used to determine the
token slot associated with RADIUS authentication requests from
the RADIUS server.
# -----------------------------------------------------------------
ACTIVCARD_APPLICATION: RADIUS
# -----------------------------------------------------------------
# ACTIVCARD_CHALLENGE: challenge_request_keyword
# The ActivCards support simultaneously two authentication codes
# for each given slot. One is a patented time/event synchronous mode
# in which the user just types the one-time password displayed by his token
# instead of the static vulnerable password he was used to. The other is
# the standard X9.9, challenge/response mode.
# ActivCard's users can choose which mode they want to use by
typing a keyword at the RADIUS password prompt:
#    login: sam
#    password: challenge_request_keyword
# Upon reception of this keyword the ActivCard component embedded in the
# RADIUS server will switch to challenge/response mode and issue a
# challenge (a very good quality random number), and the user will be
# prompted for a dynamic password:
#    login: sam
#    password: challenge_request_keyword
#
# Challenge/Response Authentication requested...
# Challenge: 12345678
# Response:
# The user has to type the challenge into his token and type at the prompt
# the dynamic password produced by the token for that challenge.
# -----------------------------------------------------------------
ACTIVCARD_CHALLENGE: challenge
# -----------------------------------------------------------------
# ACTIVCARD_HOST: 192.168.15.60
# The following parameter indicates the ip address of the machine where the
# ActivEngine is located.
# -----------------------------------------------------------------
ACTIVCARD_HOST: 192.168.15.60
Troubleshooting ActivCard

Refer to your ActivCard manuals for information on troubleshooting ActivCard. If you still have problems after trying these solutions, see Appendix E, “Contact Information for Third-Party Products,” for information on how to contact ActivCard, Inc.
Security Dynamics Technologies, Inc. provides an additional level of security in user identification and authentication by using SecurID tokens to generate codes and ACE/Server software to process the codes. This software and hardware authentication system is often referred to as SecurID.

This chapter is an overview of the installation and basic configuration of ACE/Server and ACE/Client software when used with RADIUS. This chapter includes the following topics:

- “Overview of SecurID Components” on page 7-2
- “How SecurID Works with RADIUS” on page 7-3
- “ACE/Server Installation on a UNIX Host” on page 7-4
- “RADIUS Configuration for SecurID” on page 7-10
- “Troubleshooting SecurID” on page 7-13

This information is intended to serve as a quick reference guide for the ACE/Server and ACE/Client software. Certain terms used in this chapter—such as tokencode, PASSCODE, master server, and slave server—are taken from the Security Dynamics documentation. Refer to the Security Dynamics manual set for detailed features of SecurID and future ACE/Server software releases.

Note – Lucent Remote Access Technical Support does not provide support for the ACE/Server and ACE/Client installation and configuration. See Appendix E, “Contact Information for Third-Party Products,” for information on how to contact Security Dynamics, Inc. Lucent Remote Access Technical Support provides support for RADIUS when used with ACE/Server and SecurID only after you have verified that the ACE/Server is working properly.

The ACE/Server and ACE/Client software version 2.3 is supported on the following platforms:

- AIX 4.1 on a PowerPC or a RISC/6000
- HP-UX 10.01 on a Hewlett-Packard HP 9000 Series 7xx or 8xx
Overview of SecurID Components

- Solaris 2.5.1 on a Sun SPARCstation
- SunOS 4.1.4 on a Sun SPARCstation

Additionally, the ACE/Server software is supported on Windows NT platforms. RADIUS server 2.1 is currently available only on UNIX platforms.

Overview of SecurID Components

The Security Dynamics authentication system (generally referred to as SecurID) consists of the following components:

- ACE/Server master
  Stores usernames and serial numbers of tokens and performs calculations to verify the identity of users.
- ACE/Server slave
  Functions as a secondary server to the ACE/Server master.
- ACE/Client
  A computer or other device protected by ACE/Server. ACE/Client software must be installed on these systems.
- Token
  A device that generates a random number known as a tokencode (the software might show this as two words: token code). A new number is generated and displayed every 60 seconds. Five types of tokens are supported in SecurID: the standard SecurID card, the SecurID Key Fob, SecurID PINPAD card, SecurID Modem, and SoftID. The first three tokens are small, handheld devices.

  RADIUS has been tested with the standard SecurID card, the SecurID Key Fob, and the SecurID PINPAD card.
- PASSCODE
  A two-part password, consisting of a memorized personal identification number (PIN) followed by the current tokencode displayed on the token.
**How SecurID Works with RADIUS**

*Note* – To use RADIUS with SecurID, you typically run the ACE/Server software on the same host as the RADIUS server. If you install the ACE/Server software on a different machine, then you must install the SecurID ACE/Server slave component on the RADIUS server host. The ACE/Server slave must then reference the ACE/Server master.

When SecurID is used with RADIUS, a connection proceeds as follows:

1. A remote user initiates a connection by dialing in to the PortMaster.
2. The PortMaster prompts for the user’s username and password.
3. The user enters a username. At the password prompt, the user enters a PASSCODE (PIN followed by the currently displayed number on the token).
4. The PortMaster forwards this information to the RADIUS server for authentication.
5. The RADIUS server examines the `users` file, scanning for the appropriate username. When the profile is located, it is examined to determine the user’s authentication method.
6. When the RADIUS server discovers that the authentication method is SecurID, it forwards the username and PASSCODE to the ACE/Server for authentication.
7. The ACE/Server examines its database for the username and serial number of the user’s token. It uses the serial number to verify the PASSCODE entered by the user. It also verifies that the time on the token is synchronized with the ACE/Server.
8. The ACE/Server sends the result of the database lookup (identity verified or not verified) to the RADIUS server.
9. If the user’s identity was verified by the ACE/Server, the RADIUS server sends an access-accept message to the PortMaster along with the additional information from the RADIUS user profile. If the ACE/Server rejected the user’s PASSCODE, the RADIUS server sends an access-reject message to the PortMaster.
ACE/Server Installation on a UNIX Host

The SecurID software package consists of a number of applications and utilities. This section provides guidelines for the installation and use of version 2.3 of ACE/Server and ACE/Client, and the sdadmin utility. This is not a complete explanation of all SecurID requirements or procedures. Refer to the appropriate Security Dynamics documentation for information on other versions of ACE software.

The master ACE/Server handles the authentication requests passed on to it by the RADIUS server configured as an ACE/Client. The master and client software can be installed on the same host.

You can increase the reliability of the authentication process if you configure the ACE/Client and an ACE/Server slave on a separate host or hosts from the ACE/Server master. In the event the host for the ACE/Server master goes down, the ACE/Server slave handles the authentication requests from the ACE/Client.

These instructions cover the following:

- “Installing ACE/Server and Client Software on a UNIX Host” on page 7-6
- “Administering ACE/Server with sdadmin” on page 7-7
- “Authenticating with sdshell” on page 7-8

This is not a complete explanation of all SecurID requirements or procedures. If you are upgrading an older ACE/Server installation previous to version 2.3, you must read the ACE/Server v 2.3 for UNIX Administration Manual from Security Dynamics for instructions.

Note – Read the ACE/Server v 2.3 Installation Guide before beginning installation.

SecurID software is not shipped with the PortMaster. To order this software, see Appendix E, “Contact Information for Third-Party Products.”

Getting Started

The server on which SecurID is installed has the following requisites. See the ACE/Server v 2.3 Installation Guide for more information. These requirements are for SecurID alone, and do not reflect the requirements for RADIUS.
Physical Memory

Security Dynamics recommends 32MB of physical memory. The minimum memory feasible for SecurID is 16MB.

Disk Space

Security Dynamics recommends 1GB of disk space. The minimum feasible disk space for SecurID is 400MB. At least 20 percent of the free disk space must be reserved for ACE/Server database growth.

Hostnames

The ACE/Server’s primary hostname—bootname—must be the first name in any list of aliases for that machine if you are using a name service such as NIS or DNS.

Kernel Configuration

Your system’s kernel configuration values must be set at or above the minimums specified by Security Dynamics. Table 7-1 shows the values in /etc/system for systems running Solaris. If you are running HP-UX, AIX, or SunOS 4.1.4, refer to the ACE/Server v 2.3 Installation Guide. Refer to your operating system’s manuals for instructions on setting these values.

Table 7-1   Solaris Minimum Kernel Configuration Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>shmsys:shminfo_shmmni</td>
<td>100</td>
</tr>
<tr>
<td>shmsys:shminfo_shmseg</td>
<td>16</td>
</tr>
<tr>
<td>shmsys:shminfo_shmmax</td>
<td>4194304</td>
</tr>
<tr>
<td>semsys:seminfo_semnni</td>
<td>64</td>
</tr>
<tr>
<td>semsys:seminfo_semmsl</td>
<td>50</td>
</tr>
<tr>
<td>semsys:seminfo_semmns</td>
<td>100</td>
</tr>
<tr>
<td>semsys:seminfo_semmnu</td>
<td>100</td>
</tr>
</tbody>
</table>
ACE/Server Installation on a UNIX Host

ACE/Server Service Names and Port Numbers

The SecurID authentication service has a default name of `securid` and default port number of 5500. The SecurID master/slave communication service has a default name of `securidprop` and default port number of 5510. The `sdsetup` utility adds the SecurID UDP and TCP service names and port numbers to the `/etc/services` file.

```
securid 5500/udp #ACE/Server
securidprop 5510/tcp #ACE/Server Slave
```

NIS Map

The SecurID authentication service has a default name of `securid` and default port number of 5500. The SecurID master/slave communication service has a default name of `securidprop` and default port number of 5510. If you are using NIS or NIS+, you must add these entries to the services NIS map on your NIS master and push the maps.

```
securid 5500/udp #ACE/Server
securidprop 5510/tcp #ACE/Server Slave
```

Pushing the maps updates the database to include recently entered information. Use the `make services` command on the NIS master. For details, consult your UNIX system documentation.

Installing ACE/Server and Client Software on a UNIX Host

The RADIUS 2.x server is compatible with ACE/Server versions 2.3 and higher. To install ACE/Server and ACE/Server client, complete the following steps:

1. **Log in as root.**
2. **Read the ACE/Server tape into the `ace_install` directory of the ACE/Server machine.**
   
   ACE/Server installs its software using the `sdsetup` utility.

3. **Run `sdsetup` to install ACE/Server.**

   **Note** – The `sdsetup` utility cannot be run while the `sdconnect` process or `aceserver` daemon are running. Stop these processes before attempting to run `sdsetup`.

   ```
   ace_install/sdsetup
   ```
Several options can be used with **sdsetup**. See the *ACE/Server v 2.3 Installation Guide* for more information.

The ACE/Server software is typically installed on the same machine as the RADIUS server. To run ACE/Server on a different machine, you must configure the RADIUS server as an ACE/Server slave. See the *ACE/Server v 2.3 Installation Guide* for instructions on configuring the ACE/Server slave.

4. **Continue to install the ACE/Server client software using** **sdsetup**.

   Complete instructions are given in the *ACE/Server v 2.3 Installation Guide*.

**Administering ACE/Server with sdadmin**

ACE/Server includes the **sdadmin** administration utility. Using **sdadmin**, you can add and delete users, assign PINs and tokens, and monitor network activity. You can run **sdadmin** in GUI (the default) or character mode.

To use **sdadmin**, complete the following steps:

1. **Ensure that you are in the directory that contains the ACE/Server files**.

   By default, ACE/Server software is installed in the **/usr/ace** directory.

2. **Start the database broker (sdconnect) as root**:

   ```
   /usr/ace/sdconnect start
   ```

   To stop the database broker, use the **sdconnect stop** command.

3. **Start the ACE/Server daemon using the following command**:

   ```
   /usr/ace/aceserver start
   ```

   To stop ACE/Server, use the **aceserver stop** command.
4. Add the following lines to `/etc/rc.local` or equivalent boot file of your UNIX system.

```bash
if [ -x /usr/ace/aceserver ]; then
  /usr/ace/aceserver stop
  /usr/ace/sdconnect stop
  /usr/ace/sdconnect start
  /usr/ace/aceserver start
else
  echo "Cannot start aceserver"
fi
```

These lines automatically start the ACE/Server processes `sdconnect` and `aceserver` after the host is rebooted.

5. Run `sdadmin` in GUI or character mode.

Character mode requires the use of the `-c` option:

```
/usr/ace/sdadmin &
or
/usr/ace/sdadmin -c &
```

To run `sdadmin` in GUI mode, the host’s window environment must be an implementation of X11R5 or later. If you are running SunOS, Sun OpenWindows is an X11R4 implementation, and you must therefore install the X11R5 kit shipped with the ACE/Server software. See the ACE/Server v 2.3 Installation Guide for instructions.

6. Using the instructions in the ACE/Server v 2.3 for UNIX Administration Manual, create the client, add users to the database, activate users on the client, and assign tokens to the users.

7. Choose a method of PIN assignment using the instructions for PIN administration in the ACE/Server v 2.3 for UNIX Administration Manual.

   Note that you can assign PINs using RADIUS.

**Authenticating with sdshell**

You specify which users are required to authenticate with SecurID by modifying their entries in the `/etc/passwd` file. Change the shell specification—typically `/bin/sh` or `/bin/csh`—to `/usr/ace/prog/sdshell`. This modification is applicable to all UNIX clients.
except for AIX clients that do not use a name service such as DNS or NIS. For AIX clients that do not use a name service, substitute \texttt{sdshell\_auth} for the \texttt{sdshell} authentication shell.

You can configure PIN assignments so that users must create their own PINs, must use PINs generated by the system, or can choose whether to create a PIN or use one provided to them. The default mode is to enable the user to select either a user-created or system-generated PIN. See the PIN administration information in the \textit{ACE/Server v 2.3 for UNIX Administration Manual} for configuration instructions.

If a user has forgotten her PIN, or you believe the PIN to be compromised, you must change the PIN for that token by setting the token into New PIN mode and clearing the old PIN.

If the authentication shell has been specified, the following prompts appear on the user’s screen after the user logs in to the system:

\begin{verbatim}
Enter PASSCODE:

Press <Return> to generate a new PIN and display it on the screen, or
Ctrl d to leave your token in New PIN mode:
ARE YOU PREPARED TO HAVE THE SYSTEM GENERATE A PIN? (y or n) [n]: y

Your screen will automatically clear in 10 seconds.
Your new PIN: XXXX

Wait for the code on your token to change, then log in with the new PIN
Enter PASSCODE:
PASSCODE Accepted
\end{verbatim}

The authentication shell instructs the user to enter a new PIN or press \texttt{Return} to have a PIN automatically generated. In this example, the user has a PIN generated for her.

If the user’s new PASSCODE is accepted, communication between the ACE/Server client and server is successful.

\textbf{Note} -- Lucent Remote Access Technical Support does not provide support for the ACE/Server and ACE/Client installation and configuration. See Appendix E, “Contact Information for Third-Party Products,” for information on how to contact Security Dynamics, Inc. Lucent Remote Access Technical Support provides support for RADIUS when used with ACE/Server and SecurID only after you have verified that the ACE/Server is working properly.
### RADIUS Configuration for SecurID

Each SecurID user must have a profile in the RADIUS users file or must use a DEFAULT profile. In the profile, the Auth-Type check item must be SecurID, as shown in the following example:

```
DEFAULT
  Auth-Type = SecurID
  Service-Type = Framed-User,
  Framed-Protocol = PPP,
  Framed-Address = 255.255.255.254,
  Framed-Routing = None,
  Framed-MTU = 1500
```

To activate and assign tokens to users authenticated with this DEFAULT profile, use the `sdadmin` utility, as discussed under “Administering ACE/Server with sdadmin” on page 7-7.

When user `bob` dials in to the PortMaster, the following prompts are displayed:

```
login: <enter username>
Password: <enter PIN number followed by a token code>
```

### PIN Assignment

When a new user is added to the ACE/Server database, a token is assigned to the user. How the authentication is completed depends on how you have specified PIN generation. You can require the ACE/Server to generate PINs for all users, you can force all users to provide their own PINs, or you can enable specified users to choose the generation method.

Users must provide their PINs in New PIN mode. You can also force other users into New PIN mode if they have forgotten their PINs or if an attacker has learned their PINs.

A user in New PIN mode can create the PIN using RADIUS when dialing in to the network. Refer to information on PIN administration in the `ACE/Server v 2.3 for UNIX Administration Manual` for more information on New PIN mode.
User-Created PIN

When a user in New PIN mode is forced to create a PIN via RADIUS, the user is prompted to enter a new PIN:

```
login: bob  
Password: xxxxx  
Enter PASSCODE: <token code>  
   Enter your new PIN, containing 4 to 8 digits, or  
   <Ctrl d> to cancel the new PIN procedure:
```

In this example, when user bob dials in to the network, he logs in with his username and UNIX password. When prompted for the PASSCODE—a PIN followed by the token code—bob enters the token code displayed on his SecurID device. The PortMaster sends an access-request to the RADIUS server. The ACE/Server searches its database and recognizes user bob as a New PIN mode user. It sends an access-challenge to the PortMaster, and bob is prompted to enter a new PIN.

After bob enters his new PIN, the RADIUS server responds with the following message:

```
Please re-enter new PIN:  
Wait for the code on your token to change, then log in with the new PIN  
Enter PASSCODE:  
PASSCODE accepted
```

User bob re-enters the new PIN. After a few seconds, the token code on his SecurID device changes. User bob enters the PIN and the token code, and is authenticated. For subsequent logins, bob enters his PIN followed by the currently displayed token code when prompted for the PASSCODE.

System-Generated PIN

When you specify that the PIN is generated by the system, the user is prompted to initiate PIN generation. The new PIN is displayed on the screen for the user to memorize.

**Note** – The system-generated PIN appears for only 10 seconds. After the PIN disappears, it cannot be viewed again.
In the following example, *keiko* logs in with her username and UNIX password for the first time. When prompted for the PASSCODE—a PIN followed by the token code—*keiko* enters the token code displayed on her SecurID device.

```
login: keiko
Password:
Enter PASSCODE: <token code>
    Press <Return> to generate a new PIN and display it on the screen,
    or
    <Ctrl d> to leave your token in New PIN mode:
    ARE YOU PREPARED TO HAVE THE SYSTEM GENERATE A PIN? (y or n)
    (n): y
```

When prompted, *keiko* indicated that she wants the system to generate her PIN. As shown in the following example, the PIN is displayed, and *keiko* is prompted to enter the new PASSCODE.

```
Your screen will automatically clear in 10 seconds.
Your new PIN: NNNNN
Wait for the code on your token to change, then log in with the new PIN
Enter PASSCODE:
PASSCODE Accepted.
```

For subsequent logins, *keiko* enters her system-generated PIN followed by the currently displayed token code when prompted for the PASSCODE.

### Entering an Invalid Token Code

If a user enters a valid PIN and an invalid token code, the token goes into Next Tokencode mode. The user is prompted to enter the next code from the token. This prompt also appears if the user’s token is not synchronized with the ACE/Server.

The user must wait until the token code changes and then enter the new token code number at the prompt. After the system verifies the second token code, the user is authenticated.

If an unauthorized user enters a stolen PIN followed by a guessed token code, the person is given three opportunities to enter the correct token code. If three invalid token codes are entered, the unauthorized user is disconnected.
In the following example, paolo has entered a valid PIN followed by an invalid token code. The prompt appears, indicating that paolo’s token is not synchronized with the ACE/Server, or that paolo has entered an invalid token code. User paolo must wait for 60 seconds for a new token code and then must enter this code at the prompt. In this example, paolo has entered the next code and it has been accepted.

```
login: paolo
Password: <PIN number followed by invalid token code>
Please Enter the Next Code from Your Token: <PIN number followed by next valid token code>
PASCODE Accepted
```

**Troubleshooting SecurID**

Refer to your SecurID manuals for information on troubleshooting SecurID. If you still have problems after trying these solutions, see Appendix E, “Contact Information for Third-Party Products,” for information on how to contact Security Dynamics, Inc.
RADIUS 2.1 is strictly compliant with RFC 2139. RADIUS accounting logs information about dial-in connections. This information is often used for billing purposes. RADIUS accounting consists of a client/server format. On a UNIX host, transactions are recorded as they occur in a file on the RADIUS accounting server named 

```
/usr/adm/radacct/portmastername/detail
```

This chapter includes the following topics:

- “How RADIUS Accounting Works” on page 8-1
- “Getting Started” on page 8-3
- “Client Configuration” on page 8-4
- “Accounting Server Configuration” on page 8-4
- “Accounting Attributes” on page 8-6
- “Start and Stop Records” on page 8-11

If you want to use RADIUS accounting, RADIUS 2.1 must be run with ComOS 3.3.1 or later, or with the -o option.

### How RADIUS Accounting Works

RADIUS accounting consists of an accounting server and accounting clients (PortMaster products). RADIUS accounting starts automatically when the RADIUS server starts. On a UNIX host, the `radiusd` accounting daemon is a child process of the `radiusd` authentication daemon.

The RADIUS accounting server uses the User Datagram Protocol (UDP), and listens for UDP packets at port 1646 by default.

RADIUS accounting consists of the following steps:

1. The PortMaster (accounting client) sends an `accounting-request` packet containing the record of an event to the accounting server. The record is described by the values of RADIUS attributes included in the packet.
For example, when a user is authenticated and connected, the Acct-Status-Type attribute has a value indicating that the request marks the beginning of user service. The RADIUS accounting server logs this event as a start accounting record. The records are recorded in a file called /usr/adm/radacct/portmastername/detail on the UNIX host.

Note – RADIUS automatically creates the portmastername directory and the detail file. If the IP address of a PortMaster client cannot be resolved to a hostname, then the name of the directory is the IP address of the PortMaster rather than its name.

When the user’s connection ends, the Acct-Status-Type attribute has a value indicating that the request marks the end of user service. The RADIUS accounting server records this as a stop accounting record. The stop record contains all the information in the start record plus additional information that describes what occurred during that session, such as Acct-Session-Time.

2. The accounting server sends an accounting-response packet back to the PortMaster to acknowledge receipt of the request. The server must send back an acknowledgment when it records the request.

3. If the PortMaster does not receive a response, it continues to send accounting-requests until it receives a response.

   A backoff algorithm is used to determine the delay between accounting-requests if an accounting-response is not received.

4. The PortMaster records the number of seconds that have passed between the event and the current attempt to send the record; this number is the Acct-Delay-Time value. As additional time passes before an accounting-response is received, the Acct-Delay-Time is updated.
Table 8-1 lists RADIUS attributes that a PortMaster might send to a RADIUS accounting server. See RFC 2139 for a complete list of accounting attributes. In addition, the RADIUS accounting server includes a timestamp for each entry. Information on the request-authenticator for the accounting-request packet is included if `radiusd` was run with the `-o` flag.

Table 8-1 Common Attributes in `detail` File

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-Name</td>
<td>Called-Station-Id</td>
</tr>
<tr>
<td>NAS-IP-Address</td>
<td>Calling-Station-Id</td>
</tr>
<tr>
<td>NAS-Port</td>
<td>Acct-Status-Type</td>
</tr>
<tr>
<td>Service-Type</td>
<td>Acct-Delay-Time</td>
</tr>
<tr>
<td>Framed-Protocol</td>
<td>Acct-Input-Octets</td>
</tr>
<tr>
<td>Framed-IP-Address</td>
<td>Acct-Output-Octets</td>
</tr>
<tr>
<td>Filter-Id</td>
<td>Acct-Session-Id</td>
</tr>
<tr>
<td>Login-IP-Host</td>
<td>Acct-Authentic</td>
</tr>
<tr>
<td>Login-Service</td>
<td>Acct-Session-Time</td>
</tr>
<tr>
<td>Login-TCP-Port</td>
<td>Acct-Terminate-Cause</td>
</tr>
<tr>
<td>Framed-IPX-Network</td>
<td>NAS-Port-Type</td>
</tr>
<tr>
<td>LE-Advice-of-Charge</td>
<td>Connect-Info</td>
</tr>
<tr>
<td>LE-Terminate-Detail</td>
<td></td>
</tr>
</tbody>
</table>

**Getting Started**

Select a host to use as the RADIUS accounting server. This host can be either the same host as the RADIUS server used for authentication or a separate host.

Choose a host with the following characteristics:

- Secure physical location
- Root access limited to the security officer or system administrator
- Limited number of user accounts—preferably none
Client Configuration

- Basic memory
- Enough disk space to store the RADIUS accounting detail files

RADIUS accounting data continues to grow unless you archive this information on a regular schedule—weekly or monthly, for example. For typical installations, allocate 50MB per 1000 users per month. If you archive accounting records on the server, you must allocate more storage than this minimum. Keep in mind that allocating too much space is preferable to allocating too little; your usage can vary.

For example, if you have 1000 users, one port for every 10 users, an average connection time per user of 1 hour, and all ports in use around the clock, one month of logs would require 50MB of disk space:

700 bytes/session * 1000 users * 1 port/10 users * 1 session/hour * 24 hours/day * 31 days/month

The use of a secondary RADIUS accounting server is recommended. The primary accounting server is always used first; if this server is unavailable, the secondary server is used.

The PortMaster always sends accounting packets to the primary RADIUS accounting server first and retries it once every 45 seconds. If the primary server does not respond within 10 minutes, or if there are more than 50 accounting packets waiting to be sent, the PortMaster sends the accounting packets to the secondary RADIUS accounting server. This behavior is subject to change in future releases of ComOS.

Client Configuration

To configure RADIUS accounting information on a PortMaster, see Chapter 3, “Adding a RADIUS Client.”

Accounting Server Configuration

If you have already installed the RADIUS server (radiusd) on a host, that server also acts as an accounting server. No further installation is needed.

Installation

To install the RADIUS accounting server on a UNIX host, perform the following steps:

1. Log in to the selected accounting server as root.
2. Create a `radacct` directory within the `/usr/adm` directory and grant full access only to root users:

   ```
   mkdir /usr/adm/radacct
   chmod 700 /usr/adm/radacct
   ```

   RADIUS accounting automatically creates a subdirectory within the `/usr/adm/radacct` directory for each PortMaster serving as a RADIUS accounting client and logs the accounting start and stop records to the `detail` file in the directory.

**Configuring Options**

Table 8-2 describes the `radiusd` options you can use to modify RADIUS accounting on a UNIX host.

**Table 8-2  radiusd Accounting Options**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Specifies an alternate directory for RADIUS accounting logs. The default directory is <code>/usr/adm/radacct</code>.</td>
</tr>
<tr>
<td>-o</td>
<td>Enables the RADIUS server to accept accounting packets from RADIUS clients that do not sign the Request-Authenticator according to RFC 2139. With this option, unsigned accounting records are logged and flagged with Request-Authenticator = None. Without this option, accounting packets with an unsigned Request-Authenticator (all zeros) are discarded as invalid. Use this option only if you are using RADIUS accounting details from a PortMaster running ComOS 3.3 or earlier.</td>
</tr>
<tr>
<td>-p</td>
<td>Overrides the default UDP port used by the RADIUS authentication server and the nondefault port if it is specified in the <code>/etc/services</code> file. The accounting server uses the next higher port. If you specify <code>radiusd -p 1812</code>, authentication uses port 1812 and accounting uses port 1813. The default UDP ports are 1645 for authentication and 1646 for accounting. You must configure your PortMaster to use the same ports specified with this option.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays the RADIUS version number without starting the <code>radiusd</code> daemon. This flag also applies to the RADIUS authentication server; the RADIUS authentication and accounting servers have the same version number.</td>
</tr>
</tbody>
</table>
Accounting Attributes

For RADIUS accounting to function, a series of accounting attributes are defined in the dictionary file on the RADIUS server and appear in the start and stop accounting records. Use the following descriptions of common accounting attributes to help you interpret start and stop records. Refer to RFC 2139 for information on other accounting attributes.

**Acct-Authentic**

Acct-Authentic records whether the user was authenticated by RADIUS or by the PortMaster user table. Accounting records are not generated for passthrough users, because those users are authenticated by the destination host.

**Acct-Delay-Time**

The PortMaster records the number of seconds that have passed between the event and the current attempt to send the record; this number is the Acct-Delay-Time value.

You can determine the approximate time of an event by subtracting the Acct-Delay-Time value from the time of the record’s arrival on the RADIUS accounting server.

**Acct-Input-Octets and Acct-Output-Octets**

Acct-Input-Octets records the number of bytes received from the user and Acct-Output-Octets records the number sent to the user during a session. These values appear only in stop records.

**Acct-Session-Id**

Acct-Session-Id is a unique number assigned to each start and stop record to make it easy to match the start and stop records in a detail file, and to eliminate duplicate records.

The Acct-Session-Id is a string consisting of eight uppercase hexadecimal digits. The first two digits (nn) increment each time the PortMaster is rebooted. The next six digits begin at nn000000 for the first user login after a reboot—and increment up to approximately 16 million logins. This value equals the number of logins made in one year if users log
in once a minute to every port of a 30-port PortMaster. The Acct-Session-Id appears inside double quotation marks. This format is subject to change in future releases of ComOS.

**Acct-Session-Time**

Acct-Session-Time records the user’s connection time in seconds. This information is included only in stop records.

**Acct-Status-Type**

Acct-Status-Type has two values: **Start** and **Stop**. A start record is created when a user session begins. A stop record is recorded when the session ends.

**Acct-Terminate-Cause**

The values returned by Acct-Terminate-Cause, which are shown in Table 8-3, indicate the cause of a session’s termination. This information appears only in stop records. A NAS is a network access server, such as a PortMaster.

**Table 8-3  Session Termination Causes**

<table>
<thead>
<tr>
<th>Termination Cause</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin-Reboot</td>
<td>System administrator is ending service on the NAS—for example, prior to rebooting the NAS.</td>
</tr>
<tr>
<td>Admin-Reset</td>
<td>Port was reset by an administrator.</td>
</tr>
<tr>
<td>Callback</td>
<td>Callback user was disconnected so port can be used to call the user back.</td>
</tr>
<tr>
<td>Host-Request</td>
<td>Session was disconnected or logged out by the Login-IP-Host. This attribute value can indicate normal termination of a login session, or that the remote host has failed or become unreachable.</td>
</tr>
<tr>
<td>Idle-Timeout</td>
<td>Idle timer expired for user or port.</td>
</tr>
</tbody>
</table>
Lost-Carrier | Session terminated when the modem dropped the Data Carrier Detect (DCD) signal. This value can indicate any of the following:
- The user or his modem hung up the telephone from their end; no problem exists.
- The line was dropped.
- The modem was unable to recover from severe line noise.
- The local modem dropped DCD for some other reason.

Lost-Service | Service can no longer be provided—for example, the user’s connection to a host was interrupted.

NAS-Error | NAS detected some error other than on the port, which required ending the session.

NAS-Reboot | NAS ended the session to perform a nonadministrative reboot—a system crash.

NAS-Request | NAS ended the session for a nonerror reason not otherwise listed here.

Port-Error | NAS had to reset the port. This error commonly occurs when a device attached to the port causes too many interrupts.

Port-Preempted | NAS ended the session in order to allocate the port to a higher priority use.

Port-Suspended | NAS ended the session to suspend a virtual session.

Port-Unneeded | NAS ended the session because resource usage fell below low-water mark—for example, if a bandwidth-on-demand algorithm decided that the port was no longer needed.

Service-Unavailable | NAS was unable to provide the requested service.

Session-Timeout | Session timer expired for the user.
Implementing RADIUS Accounting

**Accounting Attributes**

Table 8-3  Session Termination Causes(Continued)

<table>
<thead>
<tr>
<th>Termination Cause</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-Error</td>
<td>Because the NAS received a PPP configuration request or acknowledgment when a session was already established, it terminated the session. This error is caused by a PPP implementation error in the dial-in client.</td>
</tr>
<tr>
<td>User-Request</td>
<td>Dial-in PPP client requested that the NAS terminate the connection. This message is expected from a proper PPP client termination.</td>
</tr>
</tbody>
</table>

**Timestamp**

Timestamp records the time of arrival on the RADIUS accounting host measured in seconds since the epoch (00:00 January 1, 1970 GMT). This attribute provides a machine-friendly version of the logging time at the beginning of the accounting record. To find the actual time of the event, subtract Acct-Delay-Time from Timestamp.

**Called-Station-Id and Calling-Station-Id**

Called-Station-Id records the telephone number called by the user. Calling-Station-Id records the number the user is called from. This information is recorded when the NAS-Port-Type is ISDN, ISDN-V120, or ISDN-V110 where supported by the local telephone company. On the PortMaster 3 and the PortMaster 4, this information is available for asynchronous calls as well, where supported by the local telephone company.

**LE-Advice of Charge and LE-Terminate-Detail**

The LE-Advice-of-Charge value is a vendor-specific attribute included in RADIUS accounting stop records generated by ComOS versions 3.8 or later. This string provides any advice-of-charge information passed along by the telephone company on the ISDN D channel.

The LE-Terminate-Detail value is a vendor-specific attribute included in RADIUS accounting stop records generated by ComOS versions 3.8 or later. This string provides a detailed description of the reason the session terminated.
The RADIUS 2.1 dictionary file uses the following syntax to define vendor-specific attributes that conform to RFC 2138:

```
# Vendor-Specific attributes use the SMI Network Management Private
# Enterprise Code from the "Assigned Numbers" RFC
#
VENDOR Livingston 307

# Livingston Vendor-Specific Attributes (requires ComOS 3.8 and RADIUS 2.1)
ATTRIBUTE LE-Terminate-Detail 2 string Livingston
ATTRIBUTE LE-Advice-of-Charge 3 string Livingston
```

**NAS-Port-Type**

NAS-Port-Type records the type of port used in the connection. The port type can be any of the following: Async, Sync, ISDN, ISDN-V120, or ISDN-V110.

**Request-Authenticator**

The Request-Authenticator attribute appears in an accounting record only when the RADIUS server detects a problem with the accounting request’s digital signature. A Request-Authenticator of **None** means that the accounting request was not digitally signed and was probably sent by a PortMaster that did not sign the accounting packets because it is running ComOS 3.3 or earlier. In RADIUS for UNIX 2.0 and 2.0.1, if the value for Request-Authenticator is **Unverified**, the accounting request signature did not match the expected value. Ensure that the shared secret on the PortMaster matches the shared secret in the `/etc/raddb/clients` file.

The RADIUS 2.1 server discards unsigned accounting packets—packets with invalid request authenticator attributes—and logs an error message. The following example shows a message resulting from a request on port 1025 from a PortMaster with an IP address of 192.168.1.1:

```
accounting: client 192.168.1.1/1025 sent accounting-request with invalid request authenticator
```
You can instruct the server to accept unsigned accounting request packets by running `radiusd -o`. With this option, invalid—unsigned—accounting records are logged and flagged with `Request-Authenticator = None`.

**Start and Stop Records**

**Example 1**

The following code sample is an example start record in a PortMaster detail file.

```
Tue Jul 30 14:48:18 1996
   Acct-Session-Id = "AC000004"
   User-Name = "jaime"
   NAS-IP-Address = 172.16.64.91
   NAS-Port = 1
   NAS-Port-Type = Async
   Acct-Status-Type = Start
   Acct-Authentic = RADIUS
   Service-Type = Login-User
   Login-Service = Telnet
   Login-IP-Host = 172.16.64.25
   Acct-Delay-Time = 0
   Timestamp = 838763298
```

The `Acct-Status-Type` attribute in the record indicates whether the record was sent when the connection began (Start) or when it ended (Stop). The `Acct-Session-Id` is listed at the beginning of the record. Note that this value matches the `Acct-Session-Id` of the stop record on the following page, indicating that these records correspond to the same session.

`User-Name` specifies the username, in this case, `jaime`. `NAS-IP-Address` specifies the IP address of the PortMaster. `NAS-Port-Type` specifies that this is an asynchronous connection. `Acct-Authentic` specifies that `jaime` is authenticated via RADIUS. `Service-Type` and `Login-Service` specify that `jaime` is a login user using Telnet. `Login-IP-Host` specifies the host that user `jaime` logged in to.

The following code sample is an example stop record that is associated with the start record on the previous page. The `Acct-Session-Id` of the stop record matches that of the start record on the previous page, indicating that these records correspond to the same session.
In the stop accounting record, Acct-Session-Time specifies that jaime’s connection lasted 21 seconds. Acct-Input-Octets indicates that 22 bytes of incoming traffic were received; Acct-Output-Octets indicates that 187 bytes of outgoing traffic were sent.

The Acct-Terminate-Cause indicates that a Host-Request terminated the session, meaning that jaime logged off the host or that the host logged him off. The Acct-Delay-Time is 0 seconds, indicating that the RADIUS accounting server received the accounting-request on the first try.

Note – For more information on accounting attributes, see “Accounting Attributes” on page 8-6.
Example 2

The following is an example of a start record in a PortMaster detail file. The start record is for an ISDN PPP connection.

Tue Jul  8 08:44:17 1997
Acct-Session-Id = "1A00014E"
User-Name = "consolata"
NAS-IP-Address = 192.168.32.1
NAS-Port = 0
NAS-Port-Type = Async
Acct-Status-Type = Start
Acct-Authentic = RADIUS
Connect-Info = "33600 LAPM/V42BIS"
Called-Station-Id = "5557026"
Calling-Station-Id = "5105550285"
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 192.168.32.35
Acct-Delay-Time = 0
Timestamp = 868376657

The NAS-Port-Type specifies that the user consolata has an asynchronous connection. Called-Station-Id and Calling-Station-Id specify the destination and source of the call. Service-Type and Framed-Protocol indicate that user consolata is a framed user establishing the connection via PPP.

The example on the following page is the stop record associated with the start record on this page. The stop record indicates that the login time for user consolata was 67 seconds. The Acct-Input-Octets and Acct-Output-Octets indicate that the incoming traffic for this session was 5877 bytes, and outgoing traffic was 2418 bytes.
Tue Jul 8 08:45:24 1997
Acct-Session-Id = "1A00014E"
User-Name = "consolata"
NAS-IP-Address = 192.168.32.7
NAS-Port = 0
NAS-Port-Type = Async
Acct-Status-Type = Stop
Acct-Session-Time = 67
Acct-Authentic = RADIUS
Connect-Info = "33600 LAPM/V42BIS"
Acct-Input-Octets = 5877
Acct-Output-Octets = 2418
Called-Station-Id = "5557026"
Calling-Station-Id = "5105550285"
Acct-Terminate-Cause = User-Request
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 192.168.32.35
Acct-Delay-Time = 0
Timestamp = 868376724

This chapter includes the following topics:

- “How Proxy Service Works” on page 9-2
- “Servers Running Proxy Service” on page 9-5
- “Proxy Confederations” on page 9-6
- “Configuring Proxy Information on the Server” on page 9-7

RADIUS for UNIX supports proxy service. Proxy service enables a RADIUS for UNIX server—the proxy server—to forward an authentication request from a network access server (NAS) to a remote RADIUS server and return the remote server’s reply to the NAS. A common use for proxy service is roaming. Roaming permits two or more Internet service providers (ISPs) to allow each other’s users to dial in to either ISP’s network for service. Users traveling outside the area of one ISP’s coverage can access their services through another ISP.

Proxy service also enables an ISP to share its modem pool with that of neighboring ISPs. Suppose that during peak usage hours the modems of an ISP are so busy that some users cannot get through. It can establish a business arrangement with a neighboring ISP—if both are using RADIUS for UNIX, Lucent’s Port Authority RADIUS server, or any other proxy-compatible RADIUS server—so that its users can dial in to the neighbor’s modem pool.

For example, suppose you run an ISP that has such a modem sharing arrangement. When the users of the other ISP dial in to your modems, your server contacts the ISP’s remote server for authentication and authorization information. The remote server authenticates the user and sends, through your server, all the information needed to configure the user’s session on your client.

Because you are in a reciprocal arrangement with the other ISP, your users can dial in to it and be forwarded to your server. You and the other ISP can accumulate accounting records for each other’s users and determine how the services are billed.

Each ISP with which you have a business relationship can specify a server to act as a remote server for proxy service. In some geographic areas, ISPs are establishing consortia to pool modems throughout the region by using remote servers.
How Proxy Service Works

Suppose the user of another ISP dials in to your ISP. Your network access server—such as a PortMaster—sends its RADIUS access request to the forwarding server—your RADIUS server, also known as the proxy server. The RADIUS server parses the remote server’s realm from the user’s request. Your RADIUS server acts as a client to the remote server and shares a secret with it. RADIUS forwards the request to the remote server identified by the realm. The remote server sends a response (either access-accept, access-reject, or access-challenge) back to the forwarding server, which sends it back to the NAS. This process is illustrated in Figure 9-1.

Figure 9-1  How Proxy Service Works

If the access-request is accepted, the NAS and the forwarding and remote servers process an accounting-request as follows:

1. The NAS sends an accounting-request to the forwarding server.
2. The forwarding server writes the request to its accounting log.
3. The forwarding server forwards the request to the remote server.
4. The remote server logs the accounting-request and sends an accounting-response to the forwarding server.
5. The forwarding server sends the accounting response to the NAS.
How Proxy Service Works

Both your forwarding server, which receives the access request directly from the NAS, and the remote accounting server store the accounting information in the directories 
\texttt{/usr/adm/radacct/name\_or\_address/detail}. The RADIUS daemon on each server creates the \texttt{name\_or\_address} directory. The forwarding server substitutes the name of the NAS for \texttt{name\_or\_address}. If the IP address of the NAS cannot be resolved to a hostname, then the IP address of the NAS is used rather than its name. On the remote server, the name is based on the hostname (or IP address) of the forwarding server the request was received from.

\textbf{Note} – If the request is forwarded across a chain of forwarding servers, the accounting records are stored on all servers in the chain.

\textbf{Realms}. The forwarding server sends the request to the remote server specified by the authentication \textbf{realm}. There are two kinds of realms:

\begin{itemize}
  \item \textbf{A named realm} is the part of a user login following the \texttt{at} sign (@). For example:
    \begin{itemize}
      \item If \texttt{isolde@cornwall.net} is the user login, \texttt{cornwall.net} is the realm.
      \item If \texttt{sequoyah@cherokeee} is the user login, \texttt{cherokeee} is the realm.
    \end{itemize}
    A domain name is frequently used as the named realm to provide uniqueness.
  \item \textbf{A numbered realm} is a Called-Station-Id. You can establish a number for users to call if they need proxy service, and forward proxy requests based on the number called.
\end{itemize}

RADIUS searches for numbered realms first. If you are using numbered realms and the RADIUS server must respond to points of presence (POPs) from multiple area codes, you must specify the area code for each PortMaster in the \texttt{proxy} file on the RADIUS server.

Here is an example of a proxy file for POPs in two different area codes:

\begin{verbatim}
tracy.dog.net    frtp67w3g3$l1  2095559288
sanramon.dog.net xst1ru83vm7s3yhp  9255554613
\end{verbatim}

With the following example profile, user \texttt{bridget} can be authenticated when she calls in to either of the POPs whose numbers are listed in her profile as Called-Station-Ids:

\begin{verbatim}
bridget Password = "knppog8"
Service-Type = Framed-User,
Framed-Protocol = PPP,
Framed-IP-Address = 255.255.255.254,
Framed-Routing = None
\end{verbatim}
Note – Usernames with embedded @s—such as tristram@cornwall.net—are treated as proxy realms.

Note – RADIUS 2.1 currently supports the old username style, realm/user. Lucent Remote Access might not support this style in future releases and recommends that you avoid such usernames. The at sign (@) always takes precedence over the slash sign (/). As a consequence, the radiusd daemon interprets a/b@c as user a/b in the named realm c. Lucent Remote Access strongly recommends avoiding such mixed usage.

Proxy-State Attribute. When a forwarding server passes along an access request to another server, the server can add a Proxy-State attribute. The attribute must be returned unmodified in the reply packet, whether that response is an access-accept, an access-reject, or an access-challenge. The forwarding server removes the Proxy-State attribute before sending the response back to the requesting client. Because access-accept and access-reject replies are authenticated on the entire packet contents, the forwarding server must re-sign the packet. See “Request-Authenticator” on page 8-10 for information about packet signatures.

Each forwarding server in a proxy chain can add a Proxy-State attribute after those added by previous servers in the chain. No server can modify or remove any Proxy-State attribute except its own. The reply from the remote server maintains the order of the Proxy-State attributes. Each server in turn then removes the last Proxy-State added—which must be its own—and passes the remaining Proxy-State attributes along when it sends the reply back down the proxy chain to the NAS.

You can include the old keyword in the proxy file entry for a remote server to communicate with RADIUS servers that do not support proxy. Using this keyword prevents the server from adding a Proxy-State attribute when it forwards a request and from removing a Proxy-State attribute when it receives the reply.

A summary of the Proxy-State Attribute format is shown below.

Table 9-1  Proxy-State Attribute as Currently Implemented in Lucent RADIUS 2.1

<table>
<thead>
<tr>
<th>Octet</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Numeric code for attribute name. For Proxy-State, the code is 33.</td>
</tr>
<tr>
<td>1</td>
<td>Length of attribute in bytes. This value is currently 30.</td>
</tr>
<tr>
<td>2-5</td>
<td>Timestamp—in seconds since epoch—that packet reached forwarding server.</td>
</tr>
</tbody>
</table>
Table 9-1  Proxy-State Attribute as Currently Implemented in Lucent RADIUS

<table>
<thead>
<tr>
<th>Octet</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-9</td>
<td>Client IP Address from which packet was received.</td>
</tr>
<tr>
<td>10-11</td>
<td>Client source port from which packet was received.</td>
</tr>
<tr>
<td>12</td>
<td>ID of packet received from client.</td>
</tr>
<tr>
<td>13</td>
<td>Pad byte. This value is currently 0.</td>
</tr>
<tr>
<td>14-29</td>
<td>Client request authenticator.</td>
</tr>
</tbody>
</table>

Note – The Lucent RADIUS Proxy-State attribute consists of an opaque sequence of octets and is subject to change without notice. The attribute value is only meaningful to the server that sent the attribute.

Servers Running Proxy Service

Both the forwarding server and the remote server must be running RADIUS for UNIX version 2.1 or later, or Lucent’s Port Authority RADIUS server. Any third-party RADIUS-conforming proxy server can probably work as either the forwarding server or remote server, but Lucent Remote Access has not tested proxy service with other RADIUS servers to determine if they correctly implement proxy. Lucent Remote Access RADIUS versions 2.0.1 and earlier do not support proxy RADIUS. However, a remote server can use the earlier versions of Lucent RADIUS if the proxy file on the forwarding server has the **old** keyword set in its proxy file entry for the remote server.

The RADIUS server uses the following UDP ports by default:

- 1645 for RADIUS authentication
- 1646 for RADIUS accounting
- 1650 for proxy requests and responses
- 1651 for proxy requests and responses

The forwarding and remote servers can run on different operating systems. A RADIUS server can function both as a forwarding server and remote server, acting as a forwarding server for some realms and as a remote server for other realms. A remote server can in turn forward a request to another remote server.
Proxy Confederations

One forwarding server can forward to any number of other forwarding or remote servers, but only one per realm. A remote server can have any number of servers forwarding to it and can provide authentication for any number of realms.

Proxy Confederations

Proxy service requires planning and cooperation between different business entities. The typical implementation involves a confederation of businesses designating a single proxy forwarding server to serve as a clearinghouse server. Within each organization’s system, the forwarding server knows only the address of the clearinghouse server. The clearinghouse server knows the addresses of all remote servers in the confederation. Figure 9-2 shows how this typical proxy confederation works.

Figure 9-2  One Practical Implementation of Proxy Service

Suppose a NAS sends a proxy request to the RADIUS forwarding server. If the realm is not found in the proxy file, the request is forwarded to the clearinghouse server. The clearinghouse server performs the lookup for the request and forwards the request to the desired remote server. The remote server sends the appropriate information back to the clearinghouse server, which in turn passes the information to the original forwarding server.
Each RADIUS server also acts as a remote server to the clearinghouse server. The clearinghouse server functions as both a forwarding server and a remote server to the servers at the ends of the confederation.

If you are using a clearinghouse remote server, you can define it as your default by specifying the realm name DEFAULT (all uppercase letters). Proxy requests are forwarded to the DEFAULT server if the named realm has no other entry in the proxy file.

**Note** – You must ensure that the servers in a proxy system do not forward to each other, which creates a forwarding loop that passes packets back and forth between them. For example, this situation occurs if a proxy file has an incorrect entry that associates the realm of the next server in the proxy chain with the IP address of the previous server in the chain.

### Configuring Proxy Information on the Server

To use the proxy service, you must configure RADIUS as you would normally, with the following additions:

- You must create a **proxy** file on each forwarding server.
- If you are using any named realms, the remote server must also have a **proxy** file so that it can authenticate them. If you are using only numbered realms, the remote server needs no proxy file.

**Caution** – Because the **proxy** file contains the shared secrets for the proxy servers, verify that only root users have read and write access to the file.

### Components of the proxy File

You create a **proxy** file in the `/etc/raddb` directory on the forwarding server and, if necessary, on the remote server. Each entry or line in the **proxy** file describes one realm.
Here is a sample *proxy* file:

```
#remote server
#hostname or optional ports
#IP address shared secret realm or keywords
#------------------- ----------- -------- ------------------
radius.edu.net vd1k4%p67w3g&g1 edu.net
s134.net.com ru83vm7xst1shm!p 5551234 1812 1813
net54.edu.net 2hbr5$w*3m7xstt 5555624
s134.net.com x56jy76mpkst 5551134
rad.edu.com ch5#5eb716erth edu.com 1645
rad7.com.net 1x4z0Fapa3ep com.net 1645 1646 old
eg.edu.net e997asepdf1j edu.net old secure
```

An entry contains the following information, all separated by spaces or tabs:

- Hostname or IP address of the remote server for the realm—a server acting as a remote server to this server, but which might in turn forward the request.
- Secret shared between this same server and the remote server.
- Realm handled by the remote server.

You can optionally include the following additional information in a *proxy* file entry:

- UDP port number for RADIUS authentication
- UDP port number for RADIUS accounting
- One or more keywords to affect the server behavior
  - **old**—This keyword causes the server to strip the realm from the login name and not attach Proxy-State when forwarding access-requests. Use this keyword when you are forwarding requests to servers with RADIUS versions older than 2.1.

**Caution** — Use the **secure** keyword with only if you want certain users to be granted administrative privileges.

- **secure**—This keyword enables the remote server to authorize someone to log in to your NAS with administrative privileges. If this keyword is not present, an access-accept message from the forwarding server to the client that grants administrative access (either Service-Type = Administrative-User or Service-Type = NAS-Prompt-User) is sent to the client as an access-reject.
The RADIUS server generates a **syslog** message similar to the following (shown on three lines for clarity):

```
Jul 10 21:10:00 ra radius[14870]: remote server 192.168.96.6/1645.4 returned insecure service for client 172.16.3.24/1039.17, sending reject instead
```

- **ipass**—This keyword instructs the server to use the iPass protocol rather than the RADIUS protocol to communicate with the remote server. See Appendix E, “Contact Information for Third-Party Products,” for information on how to contact iPass, Inc. for more information.

You can include the optional information in any order in the **proxy** entry, after the first three mandatory fields. If you specify only a single UDP port number, the server interprets this as the RADIUS authentication port number. If you specify two UDP port numbers, the first number is interpreted as the RADIUS authentication port and the second number as the RADIUS accounting port. If you do not specify any ports in the proxy entry, the server uses its own port numbers for communication with the remote server.

If a remote server—the final server in the proxy chain—has a **proxy** file, the file must have an entry configured for each of the server’s own realms. This entry must include the following information:

- Server’s own hostname or IP address
- Unique shared secret—this secret is not configured on any other device
- Realm for which this server is authoritative

### Special Realms: DEFAULT and NOREALM

You can include entries in your **proxy** file for the special realms DEFAULT and NOREALM. The following example shows sample DEFAULT and NOREALM entries:

```
center.com.net   e199aespfdx4   DEFAULT
others.com.net   e19aepsfd9x4   NOREALM
```

Requests are forwarded to the DEFAULT server if their named realm has no other entry in the **proxy** file. You might use the DEFAULT realm to define the entry for a clearinghouse server.
When you want users that have no realm to be forwarded to a specific server, you can create a NOREALM entry for that server. Consider the situation where for performance reasons you have configured your network access servers to communicate with many forwarding servers that each have proxy files but no local users files. You can use the NOREALM entry in each of these proxy files to forward all the local users—those who log in without specifying a realm—to your full RADIUS server for authentication.

The last DEFAULT and NOREALM entries in the proxy file are the ones used. Whenever you update the proxy file, the RADIUS server reads it into memory in its entirety. The RADIUS server uses the copy in memory rather than reading the file each time it needs to access a proxy entry.

**On the Forwarding Server**

The RADIUS clients file must have an entry for the name or IP address and the shared secret of the NAS. If the forwarding server is in a chain of multiple servers, the clients file must contain the name or IP address and the shared secret of any servers for which it forwards requests.

The proxy file must have an entry for the name or IP address, shared secret, and realm of all remote RADIUS servers. The shared secret in the forwarding server’s proxy file must match the shared secret in the remote server’s clients file.

*Remote* in this instance means a server to which the forwarding server sends a request for authentication. That server might be the ultimate server in the proxy chain and process the request, or it might in turn forward the request on, until the request reaches the ultimate server in the proxy chain and is processed for authentication.

**On the Remote Server**

The clients file must contain the name or IP address and the shared secret of all forwarding servers. The shared secret must match the shared secret in each forwarding server’s proxy file.

If any named realms are used, the proxy file must contain the name or IP address of the remote server, an unused dummy secret, and the realm for which this remote server is authoritative. If only numbered realms are used, then no proxy file needs to be defined on the remote server.
Example Proxy Server Relationships and Configuration Steps

Figure 9-3 illustrates the client/server relationships in a system that includes two forwarding servers and a remote server.

Figure 9-3  Proxy Server Relationships

The NAS is a client to server #1. Server #1 is a client to server #2. Server #2 is a remote server to server #1, and a client to server #3. You must include each client’s name or IP address and shared secret in the clients file of its associated server. Each client/server pair must share a secret. You must configure each server’s proxy file to point to the next server in the chain, including the name or IP address, shared secret, and realm of the next server. The secret you configure in the proxy file must match the secret configured in the clients file of the next server in the chain.

Figure 9-4 shows how secrets are shared between devices in a proxy chain. In this example, shared secrets #2 and #6 can be identical, but need not be. Shared secrets #3 and #5 can be identical, but need not be.
Figure 9-5 provides a detailed example of the proxy relationships shown in Figure 9-3. In this example, the server xroad.net acts as a clearinghouse server for ISPs in Argentina (estancia.net) and New York (redsauce.net). The clearinghouse server proxies requests in both directions.

To properly configure the components of the proxy service shown in Figure 9-5, where each NAS is a PortMaster, the administrators must perform the steps described below. This example considers the case of a user with a home account in New York who travels to Argentina.

1. **On PortMaster pm1.estancia.net, enter the following commands to set the RADIUS authentication and accounting servers:**

   ```
   set authentic 192.168.190.21 1645
   set secret ujm49fud3$$
   set accounting 192.168.190.21 1646
   ```

2. **Determine the IP address or fully qualified domain name of each NAS and server.**

   In this example, you have the following from the viewpoint of someone dialing in to the ISP in Argentina:

   - PortMaster named pm1.estancia.net with IP address 192.168.190.20
   - Forwarding server named jorge.estancia.net with an IP address of 192.168.190.21 in the realm estancia.net
   - Clearinghouse server xroad.net with an IP address of 172.30.140.2
   - Remote server named vinnie.redsauce.net with an IP address of 172.16.240.111 in the realm redsauce.net
   - PortMaster named pm22.redsauce.net with IP address 172.16.240.110

3. **On forwarding server jorge.estancia.net, configure the following:**

   - Contents of `/etc/raddb/clients`
     ```
     pm1.estancia.net   ujm49fud3$$
     xroad.net          thx1984
     ```

   - Contents of `/etc/raddb/proxy`
     ```
     xroad.net               m72hbtr5$$w3xst      redsauce.net
     jorge.estancia.net      dummysecret        estancia.net
     ```

   DEFAULT can be substituted for the realm redsauce.net.
Configuring RADIUS Proxy Service

Figure 9-5  Proxy Server Relationships

Files on jorge.estancia.net
clients file
pm1.estancia.net  ujm49bud3$$
xroad.net  thx1984

proxy file
xroad.net  m72hbtr5$$w3xxt
jorge.estancia.net  dummysecret
estancia.net

**or DEFAULT

Files on clearinghouse server xroad.net
clients file
jorge.estancia.net  m72hbtr5$$w3xxt
vinnie.redsauce.net  kntbr352bdf

Files on vinnie.redsauce.net
clients file
xroad.net  qbfja97-8
pm22.redsauce.net  bws5629s$r53

proxy file
xroad.net  kntbr352bdf
vinnie.redsauce.net  dummysecret
redsauce.net

**or DEFAULT

Files on xroad.net

Files on estancia.net

Files on redsauce.net

11850005
4. **On clearinghouse server xroad.net, configure the following:**
   
   - Contents of `/etc/raddb/clients`
     
     jorge.estancia.net   m72hbtr5$w3xst
     vinnie.redsauce.net  kntbr352bfd
   
   - Contents of `/etc/raddb/proxy`
     
     vinnie.redsauce.net  qbfja97-8    redsauce.net
     jorge.estancia.net   thx1984       estancia.net

5. **On remote server vinnie.redsauce.net, configure the following:**
   
   - Contents of `/etc/raddb/clients`
     
     xroad.net         qbfja97-8
     pm22.redsauce.net bws5629s$r53
   
   - Contents of `/etc/raddb/proxy`
     
     xroad.net         kntbr352bfd    estancia.net
     vinnie.redsauce.net dummysecret2  redsauce.net

   DEFAULT can be substituted for the realm estancia.net.

6. **On PortMaster pm22.redsauce.net, enter the following commands to set the RADIUS authentication and accounting servers:**
   
   ```
   set authentic 172.16.240.111
   set secret bws5629s$r53
   set accounting 172.16.240.111
   ```

7. **Define the user profiles.**

   You must define a user profile in the `users` file of the remote server for each user that is to be authenticated via the remote server.

   If the user’s password is stored in the `users` file, an example user profile for marciano on vinnie.redsauce.net is the following:

   ```
   marciano                Password = "jmp$f87"
   Service-Type = Framed-User,
   Framed-Protocol = PPP,
   Framed-IP-Address = 255.255.255.254,
   Framed-Routing = None
   ```
However, if *marciano*’s password is stored in the `/etc/passwd` file, an example user profile is the following:

```
marciano        Auth-Type = System
               Service-Type = Framed-User,
               Framed-Protocol = PPP,
               Framed-IP-Address = 255.255.255.254,
               Framed-Routing = None
```

If *marciano* goes to Argentina on a business trip and dials in to the network at estancia.net, he must enter `marciano@redsauce.net` at the password login prompt.

8. **Run the** radiusd **daemon on servers jorge.estancia.net, xroad.net, and vinnie.redsauce.net.**

The RADIUS accounting records for proxy users are logged into the detail file of all the servers.
This appendix provides hints and tips for troubleshooting the RADIUS authentication server and the RADIUS accounting server.

Troubleshooting RADIUS Authentication

Most RADIUS authentication problems occur because the server or client was not configured correctly, or because a step was omitted during installation. Carefully check the instructions in Chapter 2, “Configuring a RADIUS Server,” and Chapter 3, “Adding a RADIUS Client,” to ensure that the authentication server was properly installed and configured.

If you have not solved the problem after reviewing the instructions in Chapter 2 and Chapter 3, read the troubleshooting suggestions in this section. Refer to Appendix B, “RADIUS for UNIX Error Messages,” for information on specific error messages.

Checking the radiusd Daemon

1. Use `radiusd -v` command to display the version number.
2. Verify that the `/etc/raddb` directory (or the directory you specify with the `-d` flag) contains the following files: dictionary, users, and clients.
   If you are using RADIUS menus, check the `menus` subdirectory.
3. Make sure `/etc/radiusd` is running.
4. Use `radiusd -x` to view incoming and outgoing packets from RADIUS.

Checking the PortMaster

1. Make sure that security is on for each port:
   
   Command> set all security on
   Command> save all
   Command> reset all
Troubleshooting RADIUS Authentication

When security is on, the `show S0` command displays (Security) in the Port Type field of its output.

**Note** – When you issue the `reset all` command, all connections on serial and asynchronous ports are dropped. On the PortMaster 4, the `reset all` command is issued to a slot; only active connections on that slot are dropped. For all PortMaster products, the console port is not affected by the `reset all` command.

2. **Use the** show global command to verify the following:
   - RADIUS server IP address is set on the PortMaster
   - Secondary RADIUS server does not have the same IP address as the primary server

3. **Make sure that the PortMaster can contact the RADIUS server:**
   
   Command> `ping Ipaddress`

4. **Make sure the secret set on the PortMaster with the** set secret password command matches the secret in the `/etc/raddb/clients` file on the RADIUS server.

   The PortMaster will not display the shared secret; however, you can set the secret again if you are not sure that it is set properly. If you update the shared secret, make sure to use the `save all` command to save the shared secret in the PortMaster nonvolatile memory.

**Checking `/etc/raddb/users`**

1. **Items in the user entries are case-sensitive. You must do the following:**
   a. Verify the spelling and capitalization of each line of the `users` file.
   b. Compare keywords against the `/etc/raddb/dictionary` file to ensure that they are the same.

2. **Verify that the user can authenticate with a clear text password before authenticating with Auth-Type = System or Auth-Type = SecurID.**
Host Unavailable

If a “Host Unavailable” message is displayed before a username is entered at the login prompt, then both of the following conditions exist:

• Autolog username was set on the port
• Default host is not responding to the login service. The default host is either defined for the port or defined in the PortMaster user table for the user. The login service—rlogin or telnet or in.pmd—is either defined for the port or defined in the PortMaster user table for the user.

If a “Host Unavailable” message is displayed after a username is entered at the login prompt, but before a password is entered, all of the following conditions exist:

• Username exists in the local user table
• No password is associated with the username
• Default host is not responding to the login service—rlogin, telnet, or in.pmd

If a “Host Unavailable” message is displayed after a username and password are entered at the login prompt, either of the following conditions might exist:

• Security for the port is disabled and the default host is not responding to the login service—rlogin, telnet, or in.pmd
• Security for the port is enabled and the default host is not responding to the login service—rlogin, telnet, or in.pmd

To verify that security is not enabled, enter the following command. If necessary, replace $1 with the port that you are using.

Command> show $1

If (Security) is not displayed in the Port-Type field, enter the following commands to enable security for the port:

Command> set $1 security on
Command> reset $1
Command> save all
Invalid Login after 30-Second Wait

If the PortMaster sends 10 access-requests at 3-second intervals and then displays an “Invalid Login” message, this message can indicate one of the following problems:

- **RADIUS is not running on the server.**
  Verify that `/etc/radusd` is running.

- **The RADIUS server is not defined correctly on the PortMaster.**
  Check the RADIUS server information using the following commands:

  ```
  Command> show global
  Command> show netcon
  ```

  Look for remote ports of 1645 or 1646 in the `netcon` output.

- **No entry exists for the PortMaster in the `/etc/raddb/clients` file.** Verify this condition by editing `/etc/raddb/clients` to verify the PortMaster hostname or IP address is correctly defined in the file.

- **`radiusd` responses are not getting back to the PortMaster.** Do the following:
  - Examine the routing table on the RADIUS server host.
  - Ping the PortMaster from the RADIUS server host.
  - Run `traceroute` on the PortMaster address from the RADIUS server host.
  - Run `traceroute` on the RADIUS server host from the PortMaster.

- **The PortMaster is ignoring `radiusd` responses.** This occurs when the access-accept or access-reject source address does not match the destination of the access-request packet—the IP address of either the primary or secondary RADIUS server. This is usually caused by one of the following:
  - Multiple IP addresses are assigned to a single Ethernet interface on the RADIUS server host.
  - Multiple Ethernet interfaces are enabled, and the RADIUS server is replying to a request from the PortMaster on an interface different from the interface that received the request.

You can use the PortMaster packet tracing command `ptrace` to determine where packets are being routed. See the PortMaster Troubleshooting Guide for more information.
Turning on the Debug Function

You can run `radiusd -x` to display debugging output. RADIUS version 2.1 or later also enables you to use SIGUSR signals. You can turn on RADIUS debugging by sending a SIGUSR1 signal to `radiusd`. Sending a SIGUSR2 signal to `radiusd` turns debugging off.

1. **Issue one of the following commands, depending on your version of UNIX, to determine the process ID for radiusd:**

   
   ```
   % ps ax | grep radiusd
   or
   % ps -ef | grep radiusd
   ```

   If you are running RADIUS in single-threaded mode—`radiusd -s`—this returns a single process ID. If you are not running in single-threaded mode, this returns a process ID for the parent `radiusd` authentication process and a process ID for the child `radiusd` accounting process. You can signal the authentication and accounting processes separately.

2. **Send the desired signal to the process.**

   To turn debugging on:

   ```
   Command> kill -USR1 processid
   ```

   To turn debugging off:

   ```
   Command> kill -USR2 processid
   ```

   The RADIUS server logs a short summary message of `radiusd` activity when either message is sent and when `radiusd` is exited.

   An example `syslog` message might appear as follows (shown on two lines instead of one for legibility):

   ```
   Mar 19 23:10:50 ra radius[14870]: counters 5 8 / 2 4 / accept 4 reject 1 challenge 0 response 8
   ```

   This messages shows that the RADIUS server received the following:

   - 5 packets on port 1645—unless the RADIUS port was specified differently by running `radiusd -p Portnumber`
   - 8 packets on the RADIUS accounting port
• 2 RADIUS proxy replies
• 4 RADIUS accounting proxy replies

This message also shows that the RADIUS server sent the following:
• 4 access-accept packets
• 1 access-reject
• 0 access-challenges
• 8 accounting responses

The following example log message is a summary (shown on two lines instead of one for legibility) of memory allocation for each of the four major data structures that 


```
Jul 28 09:56:01 ra radiusd[19340]: memory usage=pair 8/35/4784 peer 0/0/0 req 1/4/570 buf 1/4/570
```

The memory allocation is in the format \textit{x/y/z}, where \textit{x} is the number allocated but not yet freed, \textit{y} is the high-water mark—the most ever allocated but not freed at one time—and \textit{z} is the total number allocated.

\textbf{Result of Debug Output}

If debug output shows more than one access-reject packet sent for the same ID, check the following:

1. \textbf{Check the route back to the PortMaster; ensure that replies are getting to the PortMaster.}
2. \textbf{Check to see if the RADIUS server host has more than one Ethernet port or multiple IP addresses are assigned to the same Ethernet interface.}
3. \textbf{Check for packet filters between the RADIUS server host and the PortMaster filtering out the RADIUS return packets.}
4. \textbf{On the PortMaster, use \texttt{ptrace} as follows to show packets returning from the host running \texttt{radiusd}.}

For ComOS versions below 3.7:
```
Command> add filter r
Command> set filter r 1 permit udp src eq 1645
```
Note – ptrace on a PortMaster only shows UDP or ICMP packets generated on the PortMaster itself if the PortMaster is running ComOS version 3.7 or later. Outgoing RADIUS access requests are not shown; however, returning packets are displayed. To turn off tracing, use the ptrace command with no values. See the PortMaster Command Line Reference for more information.

For ComOS version 3.7 or higher:

Command> set filter r 2 permit icmp
Command> ptrace r

5. Check the source address of a packet during tracing.

A multihomed RADIUS host might be using the wrong source address when replying to access-request packets.

If debugging output shows an access-reject packet right away, check the following:

1. Check the spelling of the username and password.
   The capitalization must match exactly.

2. Check the log file.
   Check syslog for errors from radiusd.

3. Use the show table user command to verify that the user is not in the PortMaster user table.
   The local user table is always checked first during authentication attempts.

4. If Auth-Type = System is not working, attempt to use a clear text password in the user profile.

5. If Auth-Type = System is specified on a UNIX system that has shadow passwords, ensure that radiusd is run as root to access the shadow passwords.
6. **Verify the spelling, capitalization, and syntax of the** `/etc/raddb/users` **file.**

   If `radiusd` finds any errors in the user profile, it sends an access-reject message and logs an error to `syslog`.

7. **Check that the shared secret in** `/etc/raddb/clients` **matches the one set on the** `PortMaster` **with the** `set secret` **command**

8. **If you are using PMconsole 3.5 or earlier, ensure that the secret was not inadvertently erased.**

   Pressing the **Return** key with the cursor in the RADIUS Secret field of the RADIUS window erases the secret when the **Save** button is clicked. Lucent Remote Access recommends using PMVision.

**Performance Degradation**

If you experience a degradation in RADIUS performance, consider the following two possibilities:

- Non-RADIUS processes on the host machine might be using host resources. Review the recommendations in “Selecting a RADIUS Server Host” on page 2-1. Removing non-RADIUS services can free-up host resources.

- User lookups take more time as the number of user profiles on the RADIUS server increases. Lucent Remote Access recommends caching user profiles when the `users` file contains more than 500 users to increase the speed and efficiency of user lookups.

**Troubleshooting RADIUS Accounting**

Most RADIUS accounting problems occur because a step was skipped during installation. Carefully check the instructions in Chapter 3, “Adding a RADIUS Client,” and Chapter 8, “Implementing RADIUS Accounting,” to ensure that the accounting server was properly installed and configured.

If you have not solved the problem after reviewing the instructions in Chapter 3 and Chapter 8, do the following:

1. **Verify the RADIUS accounting service directory**, `/usr/adm/radacct`, **exists.**

2. **Verify that the account used to execute** `radiusd` **has write permission to the** RADIUS accounting service directory.
3. **Check the RADIUS version number.**
   Run `radiusd -v`.

4. **Make sure that you do not have any other process bound to UDP ports 1645 or 1646.**
   Kill `radiusd` and use the `netstat -a` command. Start `radiusd` and use the `netstat -a` command again.

   **Note** – Note that some UNIX operating systems display the sockets symbolically as `.radius` and `.radacct` rather than .1645 and .1646.

5. **Use the show global command to verify that the IP address of the accounting host has been configured on the PortMaster.**
   If it has not been configured, set it using the `set accounting Ipaddress` command on the PortMaster, where `Ipaddress` is the IP address of the host running `radiusd`.

6. **Check syslog (auth.warning) for error messages from radiusd.**
   During normal use, very few error messages should appear.

7. **Ping the PortMaster from the RADIUS server to check connectivity.**

8. **If the previous suggestions do not solve the problem, run radiusd -x on the RADIUS server host and check to determine if accounting records are displayed.**
This appendix presents error messages that can be generated by RADIUS for UNIX and explains each message. Actions you can take to correct the error are recommended wherever possible. Error messages are likely to change between releases of RADIUS for UNIX.

**Note** – If no action is recommended, do the following before you contact Lucent Remote Access Technical Support: Write down the error message exactly and record the output of `radiusd -v` and `uname -a`.

### Accounting

The following error messages are related to RADIUS accounting.

```
accounting: client Ipaddress/Portnumber sent accounting-request with invalid request authenticator
```

**Explanation**

RADIUS received an accounting packet with an invalid Request-Authenticator on port `Portnumber` from a RADIUS client at `Ipaddress`.

**Recommended Action**

Check that the shared secret on the client and the server are exactly the same. Case is significant. If the secrets match, next check the version of ComOS on the client if it is a PortMaster. Lucent Remote Access recommends that you run ComOS 3.3.1 or later with RADIUS 2.1 accounting. If any of your PortMaster clients are running ComOS 3.3 or earlier, run `radiusd` with the `-o` option. If this message results from a client that is not a PortMaster, then the client is not compliant with RFC 2139.
Accounting

```
accounting: could not append to file Filename
```

**Explanation**
RADIUS was not able to add additional information to the end of the accounting **detail** file, **Filename**.

**Recommended Action**
Check permissions and ownership of the **detail** file and its directory. Verify that the hard disk has not run out of available storage space.

```
accounting: from Client - unknown client ignored
```

**Explanation**
An accounting packet was received from a network access server (NAS) that is not identified in the **clients** file.

**Recommended Action**
Add client to **clients** file if appropriate. Otherwise, apply a filter to prevent access requests from the NAS.

```
acct bind error Error message
```

**Explanation**
At startup, **radiusd** was unable to bind RADIUS accounting on UDP socket 1646 and returned an error message.

**Recommended Action**
Verify that **radiusd** is not already running. Use **netstat -a** to determine whether UDP port 1646 is in use by some other process. Kill the process that bound the port and restart **radiusd**.
At startup, `radiusd` was unable to open RADIUS accounting UDP socket 1646 and returned an error message.

**Recommended Action**
Verify that `radiusd` is not already running. Use `netstat -a` to determine whether UDP port 1646 is in use by some other process. Kill the process that bound the port and restart `radiusd`.

**Explanation**
An operating system error prevented RADIUS from starting the accounting daemon from `radiusd`.

**Recommended Action**
Consult your system administrator or the UNIX `man` pages.

---

**Authentication**

The following error messages are related to RADIUS authentication.

**Explanation**
RADIUS did not authenticate user *Username* attempting access from server *NAS* at IP address *Ipaddress*.
Authentication

auth: access-request from NAS ignored; no user name

Explanation
RADIUS received an access request with no username from server NAS. This message can result from a malformed authentication request or from the dial-up software.

Recommended Action
Check the client.

auth: access-request from unknown client NAS ignored; user name Username

Explanation
RADIUS ignored an access-request by user Username from server NAS. Any request coming from a NAS not specifically configured in the clients file is ignored.

Recommended Action
Add the NAS hostname or IP address to the clients file if appropriate. Otherwise, apply a filter to prevent access requests from the NAS.

auth bind error Error message

Explanation
At startup, radiusd was unable to bind RADIUS authentication on UDP socket 1645 and returned an error message.

Recommended Action
Verify that radiusd is not already running. Use netstat -a to determine whether UDP port 1645 is in use by some other process. Kill the process that bound the port and restart radiusd.
**Explanation**
At startup, `radiusd` was unable to open RADIUS authentication UDP socket 1645 and returned an error message.

**Recommended Action**
Verify that `radiusd` is not already running. Use `netstat -a` to determine whether UDP port 1645 is in use by some other process. Kill the process that bound the port and restart `radiusd`.

**Explanation**
RADIUS dropped packet `Idnumber` from server NAS. Each request contains an ID number. This message is generated if RADIUS receives a request with an ID to which it has already responded in the last five seconds.

**Recommended Action**
Do the following:

- Check additional error messages containing this ID number for an explanation of why RADIUS did not respond.
- Verify that the RADIUS server is able to respond to the client within 3 seconds. If your `users` file is a flat text file, consider using `builddb` to convert it to DBM format for faster lookups. You must run `radiusd` with the `-b` option to use the DBM version of the `users` file.
- Verify that the client is represented in the `clients` file.
- Verify that the RADIUS client is configured with the same RADIUS secret that is in the `clients` file.
- Investigate whether the host is overloaded with traffic; if so, consider spreading the load across several hosts.
dropping request for id Idnumber from NAS; Number requests already in queue

**Explanation**

RADIUS is dropping the request for ID Idnumber because the queue has grown too large. Number indicates the number of requests already in the queue.

**Recommended Action**

Do the following:

- Check additional error messages containing this ID number for an explanation of why RADIUS did not respond.
- Verify that the RADIUS server is able to respond to the client within 3 seconds. If your users file is a flat text file, consider using builddbm to convert it to DBM format for faster lookups. You must run radiusd with the -b option to use the DBM version of the users file.
- Verify that the client is represented in the clients file.
- Verify that the RADIUS client is configured with the same RADIUS secret that is in the clients file.
- Investigate whether the host is overloaded with traffic; if so, consider spreading the load across several hosts.

```
radrecv: fatal system error: out of memory, exiting
```

**Explanation**

RADIUS ran out of memory while receiving a packet and shut down.
**Explanation**
RADIUS received a packet from client *Client* that was reported to be *Number* of bytes in length but was *Number2* of bytes instead.

**Recommended Action**
Check client; this message can result from a malformed or corrupted packet.

```
radrecv: request from client Client claimed length Number, only Number2 bytes found
```

**Explanation**
Child process ID *Number* was not found.

```
rad_request: child Number not found
```

**Explanation**
RADIUS dropped an authentication request with ID *Number* because it had responded to an earlier request with this ID within the last five seconds. A NAS sends duplicate requests if it does not receive a response within three seconds.

**Recommended Action**
Do the following:

- Identify the first occurrence of the ID and see why it was rejected.
- Make sure that the secret in the *clients* file for the NAS server and the RADIUS secret on the NAS server match exactly.
- Make sure that the RADIUS server is able to respond within three seconds.

```
rad_request: dropped duplicate ID Number
```
Explanation
RADIUS received a request from Ipaddress2 in response to an access-challenge sent to Ipaddress, but the addresses do not match.

```
rad_request: error: mismatched IP addresses in request
Ipaddress != Ipaddress2 for ID Number Number2
```

### Explanation
The **msgget** function for the authentication key with a hexadecimal value of Hexvalue for packet ID Number failed and returned error message Number.

```
rad_request: error: msgget for key Hexvalue for id Number
returned error Number
```

### Explanation
The **msgsend** function for the authentication key with a hexadecimal value of Hexvalue for packet ID Number failed and returned error message Number.

```
rad_request: error: msgsnd for key Hexvalue for id Number
returned error Number
```

### Explanation
RADIUS failed to get the group name Group from UNIX for user Username.

```
unix_group: getgrnam(Group) for "Username" failed
```

### Explanation
RADIUS failed to get the password name from UNIX for user Username.

```
unix_group: getpwnam for "Username" failed
```
Clients

The following error messages are related to RADIUS clients.

```plaintext
child_authenticate: msgctl for msgid Number returned error: Errormessage
```

**Explanation**
During the authentication process, the `msgctl` function for message ID `Number` returned an error message.

```plaintext
child_authenticate: msgget for key Hexvalue for id Number returned error: Errormessage
```

**Explanation**
During the authentication process, the `msgget` function for the authentication key with a hexadecimal value of `Hexvalue` for packet ID `Number` failed and returned an error message.

```plaintext
child_authenticate: msgrcv for msgid Number returned error: Errormessage
```

**Explanation**
During the authentication process, the `msgrcv` function for message ID `Number` returned an error message.

```plaintext
client Ipaddress not found in client cache
```

**Explanation**
The IP address `Ipaddress` for a client NAS was not found in the client cache that is generated from the `clients` file.

**Recommended Action**
Verify that `clients` file contains this client’s IP address.
client cache entry for Client could not be parsed

**Explanation**
A syntax error occurred in the clients file for the client Client.

could not cache client datum for host Hostname

**Explanation**
RADIUS could not resolve Hostname found in the clients file into an IP address, or there are multiple entries in the clients file for the same client.

**Recommended Action**
Remove duplicate hosts from the clients file.

Error: clients file Filename not found

**Explanation**
RADIUS could not find the clients file Filename, which is read into memory and cached whenever it changes.

Error: could not create temporary client cache file Filename

**Explanation**
RADIUS could not create a temporary client cache file Filename.

**Recommended Action**
Verify that radiusd has write permission to /etc/raddb (or the directory specified with the -d option) and that the hard disk has not run out of available storage space.
Dictionary

The following error messages are related to the RADIUS dictionary.

**Error:** could not read clients file *Filename*

*Explanation*
RADIUS cannot open the clients file *Filename* for reading.

*Recommended Action*
Check the permissions on the clients file.

**attribute has non-numeric value on line Number of dictionary *Filename***

*Explanation*
When parsing the dictionary file *Filename*, RADIUS found an attribute that did not have an associated number on line *Number* of the dictionary.

*Recommended Action*
Correct the attribute in the dictionary.

**attribute has unknown type on line Number of dictionary *Filename***

*Explanation*
When parsing the dictionary file *Filename*, RADIUS found an unrecognizable attribute on line *Number* of the dictionary.

*Recommended Action*
Correct the attribute in the dictionary.
When parsing the dictionary file `Filename`, RADIUS found an attribute name on line `Number` of the dictionary that was more that 31 characters long.

**Recommended Action**
Correct the attribute name in the dictionary.

When parsing the dictionary file `Filename`, RADIUS could not open the dictionary file `Filename` for reading.

**Recommended Action**
Check permissions for the dictionary file.

When parsing the dictionary file `Filename`, RADIUS found an invalid attribute on line `Number` of the dictionary.

**Recommended Action**
**Invalid value entry on line Number of dictionary Filename**

**Explanation**
When parsing the dictionary file *Filename*, RADIUS found an invalid value for an attribute on line *Number* of the dictionary.

**Recommended Action**

---

**Ran out of memory after reading line Number of dictionary Filename**

**Explanation**
RADIUS ran out of memory after reading line *Number* of the dictionary file *Filename*.

**Recommended Action**
Either exit any non-RADIUS processes running on the RADIUS server or upgrade server memory.

---

**Value has non-numeric value on line Number of dictionary Filename**

**Explanation**
When parsing the dictionary file *Filename*, RADIUS found a nonnumeric value on line *Number* of the dictionary.

---

**Value name too long on line Number of dictionary Filename**

**Explanation**
When parsing the dictionary file *Filename*, RADIUS found a value that was more than 31 characters long on line *Number* of the dictionary.
Menu

The following error messages are related to RADIUS menus.

| parse error for menu Filename |

**Explanation**
RADIUS had trouble analyzing the data found in the menu file `Filename`.

Miscellaneous

The following are miscellaneous error messages.

| exit on signal Number |

**Explanation**
RADIUS quit because a fatal error occurred on signal number `Number`.

| sending SIGHUP signal to unresponsive child process Number |

**Explanation**
RADIUS terminated the child process ID `Number` because the process had not exited.

**Recommended Action**
Run `radiusd -s` to force RADIUS to run in single-process mode.
setexp: system error: out of memory

**Explanation**
The operating system hosting the RADIUS server does not have enough memory for RADIUS to run.

**Recommended Action**
Either exit any non-RADIUS processes running on the RADIUS server or upgrade server memory.

system error: could not fork at startup

**Explanation**
RADIUS is unable to fork a process during startup.

**Recommended Action**
Do one of the following:

- Determine why the operating system will not let RADIUS fork.
- Use `radiusd -s` to force RADIUS to run in single-process mode.

unknown request type Number from NAS ignored

**Explanation**
RADIUS ignored a request from server NAS because it could not identify the request type `Number`.

**Recommended Action**
Check the client.
The following error messages are related to SecurID.

```
securid: cannot initialize connection to SecurID server
```

**Explanation**
RADIUS could not connect to the SecurID server.

**Recommended Action**
Do the following:
- Verify that SecurID server is functional by testing it with ClientID.
- Verify that you can ping the host running the SecurID server from the host running the RADIUS server.

```
securid: error reading sdconf.rec
```

**Explanation**
The SecurID configuration file could not be opened for reading.

```
securid: SecurID server returned unknown code Number for user Username
```

**Explanation**
The SecurID server returned an unknown code *Number* for user *Username*.

**Recommended Action**
Use the SecurID documentation to identify the unknown code.
Users

The following error messages are related to RADIUS users.

```plaintext
securid: unexpected STATE="State"
```

**Explanation**
The State attribute sent in the access-request does not match the one sent in the access-challenge. Most likely the client is doing something wrong.

**Users**

The following error messages are related to RADIUS users.

```plaintext
user_find: unable to parse check-items for user Username
```

**Explanation**
RADIUS found a syntax error in the first line of the user profile in the users file for either user Username or the user who attempted access immediately before Username.

**Recommended Action**
Do the following in the user profile for Username and the previous user:

- Verify spelling and capitalization in the first line. Attribute values and check items are case-sensitive.
- Verify that commas rather than periods separate items in the first line.
- Verify that each user profile is separated by a blank line.
- Verify that the number of characters in the first line does not exceed 255.

```plaintext
user_find: unable to parse check-items in dbm entry for user Username
```

**Explanation**
RADIUS found a syntax error in the first line of the user profile for user Username in the DBM database generated from the users file.
Explanation
RADIUS found a syntax error in the reply items of the user profile for user *Username* in the DBM database generated from the *users* file.

user_find: unable to parse reply-items for user *Username*

**Explanation**
RADIUS found a syntax error in the reply items of the user profile in the *users* file for user *Username*.

**Recommended Action**
Do the following in the reply items in the user profile for *Username*:

- Verify spelling and capitalization. Attribute values and reply item names are case-sensitive.
- Verify that each reply item line contains only one reply item.
- Verify that commas rather than periods separate items in the first line.
- Verify that each reply item line—except the last—ends in a comma.

user_find: user record for user *Username* is too big, *Number* exceeds *Max*

**Explanation**
The user profile in the *users* file for user *Username* is *Number* characters in size and exceeds the maximum number of characters *Max* allowed for the profile.
user_find: zero length username rejected

**Explanation**
RADIUS rejected a username with no characters.

user_open: could not read user dbm file *Filename*

**Explanation**
RADIUS could not open the DBM users database *Filename* for reading.

user_open: could not read user file *Filename*

**Explanation**
RADIUS could not open the **users** file *Filename* for reading.

userparse: system error: out of memory

**Explanation**
RADIUS ran out of memory while parsing the user profile.
You issue the `radiusd` command with options to perform various actions in RADIUS for UNIX. Table 3-1 describes the actions and commands.

Table 3-1  RADIUS Actions and UNIX Commands

<table>
<thead>
<tr>
<th>RADIUS Action</th>
<th>UNIX Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>To specify an alternate directory for RADIUS accounting.</td>
<td><code>radiusd -a</code> Directory</td>
</tr>
<tr>
<td>The default directory is <code>/usr/adm/radacct</code>.</td>
<td></td>
</tr>
<tr>
<td>To use the DBM version of the users file. See “Configuring Database Caching of User Profiles” on page 4-40. This option is highly recommended.</td>
<td><code>radiusd -b</code></td>
</tr>
<tr>
<td>To specify an alternate directory for RADIUS configuration files.</td>
<td><code>radius -d</code> Directory</td>
</tr>
<tr>
<td>The default directory is <code>/etc/raddb</code>.</td>
<td></td>
</tr>
<tr>
<td>To specify a password file other than <code>/etc/passwd</code>.</td>
<td><code>radiusd -f</code> Filepath</td>
</tr>
<tr>
<td>To instruct the RADIUS server to bind to the specific address <code>Ipaddress</code> to listen for requests rather than binding to any address. This option is useful for multihomed servers.</td>
<td><code>radiusd -i</code> <code>Ipaddress</code></td>
</tr>
<tr>
<td>To specify a RADIUS logfile to use instead of <code>syslog</code>.</td>
<td><code>radiusd -l</code> Filepath</td>
</tr>
<tr>
<td>To enable the RADIUS server to accept accounting packets from RADIUS clients that do not sign packets as required by RFC 2139. With this option, unsigned accounting records are logged and flagged with <code>Request-Authenticator = None</code>.</td>
<td><code>radiusd -o</code></td>
</tr>
</tbody>
</table>
RADIUS Options

To override the default UDP ports—or nondefault ports specified in the /etc/services file—used by the RADIUS authentication and accounting servers.

radiusd -p Portnumber

For backwards compatibility, the default UDP ports are 1645 for authentication, 1646 for accounting, 1650 and 1651 for proxy. Port 1812 is reserved for RADIUS authentication. Port 1813 is reserved for RADIUS accounting.

For example, if you specify radiusd -p 1812, the following results:

- The authentication server uses port 1812
- The accounting server uses port 1813
- Proxy requests and responses are handled on ports 1817 and 1818.

To run RADIUS in single-threaded mode without spawning a child process to handle each authentication request.

radiusd -s

To display the version of RADIUS without starting the radiusd daemon.

radiusd -v

To enable debug mode.

radiusd -x

To send debug output to syslog.

radiusd -x -l syslog.

Table 3-1 RADIUS Actions and UNIX Commands (Continued)

<table>
<thead>
<tr>
<th>RADIUS Action</th>
<th>UNIX Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>To override the default UDP ports—or nondefault ports specified in the /etc/services file—used by the RADIUS authentication and accounting servers.</td>
<td>radiusd -p Portnumber</td>
</tr>
<tr>
<td>The following results:</td>
<td></td>
</tr>
<tr>
<td>- The authentication server uses port 1812</td>
<td></td>
</tr>
<tr>
<td>- The accounting server uses port 1813</td>
<td></td>
</tr>
<tr>
<td>- Proxy requests and responses are handled on ports 1817 and 1818.</td>
<td></td>
</tr>
</tbody>
</table>

To run RADIUS in single-threaded mode without spawning a child process to handle each authentication request.

radiusd -s

To display the version of RADIUS without starting the radiusd daemon.

radiusd -v

To enable debug mode.

radiusd -x

To send debug output to syslog.

radiusd -x -l syslog.
This appendix lists the RADIUS dictionary, version 1.7. Always use the latest RADIUS dictionary available at ftp://ftp.livingston.com/pub/le/radius/dictionary.

#---------------------------------------------------------------------------------
# @(#)dictionary1. 7 6/1/98 Copyright 1991-1998 Lucent Technologies, Inc
# #---------------------------------------------------------------------------------
# This file contains dictionary translations for parsing
# requests and generating responses. All transactions are
# composed of Attribute/Value Pairs. The value of each attribute
# is specified as one of 4 data types. Valid data types are:
# string - 0-253 octets
# ipaddr - 4 octets in network byte order
# integer - 32 bit value in big endian order (high byte first)
# date - 32 bit value in big endian order - seconds since
# 00:00:00 GMT, Jan. 1, 1970
# Enumerated values are stored in the user file with dictionary
# VALUE translations for easy administration.
# Example:
# ATTRIBUTE VALUE
# ------------------- ---------
# Framed-Protocol = PPP
# 7 = 1 (integer encoding)
#
# Obsolete names for backwards compatibility with older users files
# If you want RADIUS accounting logs to use these obsolete names
# instead of the current ones, move this section to the end of the
# dictionary file and kill and restart radiusd
# If you don't have a RADIUS 1.16 users file that you're still using,
# you can delete or ignore this section.
#
ATTRIBUTE Client-Id 4 ipaddr
ATTRIBUTE Client-Port-Id 5 integer
ATTRIBUTE User-Service-Type 6 integer
ATTRIBUTE Framed-Address 8 ipaddr
ATTRIBUTE Framed-Netmask 9 ipaddr
ATTRIBUTE Framed-Filter-Id 11 string
ATTRIBUTE Login-Host 14 ipaddr
ATTRIBUTE Login-Port 16 integer
ATTRIBUTE Old-Password 17 string
ATTRIBUTE Port-Message 18 string
ATTRIBUTE Dialback-No 19 string
ATTRIBUTE Dialback-Name 20 string
ATTRIBUTE Challenge-State 24 string
VALUE Service-Type Dialback-Login-User 3
VALUE Service-Type Dialback-Framed-User 4
VALUE Service-Type Shell-User 6
VALUE Framed-Compression Van-Jacobsen-TCP-IP 1
VALUE Auth-Type Unix 1
#
# END of obsolete names for backwards compatibility
#
#
# Vendor-Specific attributes use the SMI Network Management Private
# Enterprise Code from the "Assigned Numbers" RFC
#
VENDOR Livingston 307
#
# Livingston Vendor-Specific Attributes (requires ComOS 3.8 and RADIUS 2.1)
#
ATTRIBUTE LE-Terminate-Detail 2 string Livingston
ATTRIBUTE LE-Advice-of-Charge 3 string Livingston
#
# Current names for attributes
#
ATTRIBUTE User-Name 1 string
<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>2</td>
<td>string</td>
</tr>
<tr>
<td>CHAP-Password</td>
<td>3</td>
<td>string</td>
</tr>
<tr>
<td>NAS-IP-Address</td>
<td>4</td>
<td>ipaddr</td>
</tr>
<tr>
<td>NAS-Port</td>
<td>5</td>
<td>integer</td>
</tr>
<tr>
<td>Service-Type</td>
<td>6</td>
<td>integer</td>
</tr>
<tr>
<td>Framed-Protocol</td>
<td>7</td>
<td>integer</td>
</tr>
<tr>
<td>Framed-IP-Address</td>
<td>8</td>
<td>ipaddr</td>
</tr>
<tr>
<td>Framed-IP-Netmask</td>
<td>9</td>
<td>ipaddr</td>
</tr>
<tr>
<td>Framed-Routing</td>
<td>10</td>
<td>integer</td>
</tr>
<tr>
<td>Filter-Id</td>
<td>11</td>
<td>string</td>
</tr>
<tr>
<td>Framed-MTU</td>
<td>12</td>
<td>integer</td>
</tr>
<tr>
<td>Framed-Compression</td>
<td>13</td>
<td>integer</td>
</tr>
<tr>
<td>Login-IP-Host</td>
<td>14</td>
<td>ipaddr</td>
</tr>
<tr>
<td>Login-Service</td>
<td>15</td>
<td>integer</td>
</tr>
<tr>
<td>Login-TCP-Port</td>
<td>16</td>
<td>integer</td>
</tr>
<tr>
<td>Reply-Message</td>
<td>18</td>
<td>string</td>
</tr>
<tr>
<td>Callback-Number</td>
<td>19</td>
<td>string</td>
</tr>
<tr>
<td>Callback-Id</td>
<td>20</td>
<td>string</td>
</tr>
<tr>
<td>Framed-Route</td>
<td>22</td>
<td>string</td>
</tr>
<tr>
<td>Framed-IPX-Network</td>
<td>23</td>
<td>ipaddr</td>
</tr>
<tr>
<td>State</td>
<td>24</td>
<td>string</td>
</tr>
<tr>
<td>Class</td>
<td>25</td>
<td>string</td>
</tr>
<tr>
<td>Vendor-Specific</td>
<td>26</td>
<td>string</td>
</tr>
<tr>
<td>Session-Timeout</td>
<td>27</td>
<td>integer</td>
</tr>
<tr>
<td>Idle-Timeout</td>
<td>28</td>
<td>integer</td>
</tr>
<tr>
<td>Termination-Action</td>
<td>29</td>
<td>integer</td>
</tr>
<tr>
<td>Called-Station-Id</td>
<td>30</td>
<td>string</td>
</tr>
<tr>
<td>Calling-Station-Id</td>
<td>31</td>
<td>string</td>
</tr>
<tr>
<td>NAS-Identifier</td>
<td>32</td>
<td>string</td>
</tr>
<tr>
<td>Proxy-State</td>
<td>33</td>
<td>string</td>
</tr>
<tr>
<td>Acct-Status-Type</td>
<td>40</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Delay-Time</td>
<td>41</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Input-Octets</td>
<td>42</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Output-Octets</td>
<td>43</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Session-Id</td>
<td>44</td>
<td>string</td>
</tr>
<tr>
<td>Acct-Authenticator</td>
<td>45</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Session-Time</td>
<td>46</td>
<td>integer</td>
</tr>
<tr>
<td>Acct-Terminate-Cause</td>
<td>49</td>
<td>integer</td>
</tr>
<tr>
<td>NAS-Port-Type</td>
<td>61</td>
<td>integer</td>
</tr>
<tr>
<td>Port-Limit</td>
<td>62</td>
<td>integer</td>
</tr>
<tr>
<td>Tunnel-Type</td>
<td>64</td>
<td>integer</td>
</tr>
<tr>
<td>Tunnel-Medium-Type</td>
<td>65</td>
<td>integer</td>
</tr>
</tbody>
</table>
### RADIUS Dictionary

**ATTRIBUTE** Tunnel-Server-Endpoint 67 string

**ATTRIBUTE** Connect-Info 77 string

#
# Non-Protocol Attributes
# These attributes are used internally by the server
#
**ATTRIBUTE** Expiration 21 date
**ATTRIBUTE** Auth-Type 1000 integer
**ATTRIBUTE** Menu 1001 string
**ATTRIBUTE** Termination-Menu 1002 string
**ATTRIBUTE** Prefix 1003 string
**ATTRIBUTE** Suffix 1004 string
**ATTRIBUTE** Group 1005 string
**ATTRIBUTE** Crypt-Password 1006 string
**ATTRIBUTE** Connect-Rate 1007 integer

#
# Integer Translations
#
# User Types

<table>
<thead>
<tr>
<th>VALUE</th>
<th>Service-Type</th>
<th>Login-User</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Framed-User</td>
<td>2</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Callback-Login-User</td>
<td>3</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Callback-Framed-User</td>
<td>4</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Outbound-User</td>
<td>5</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Administrative-User</td>
<td>6</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>NAS-Prompt-User</td>
<td>7</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Call-Check</td>
<td>10</td>
</tr>
<tr>
<td>VALUE</td>
<td>Service-Type</td>
<td>Call-Check-User</td>
<td>129</td>
</tr>
</tbody>
</table>

# Framed Protocols

<table>
<thead>
<tr>
<th>VALUE</th>
<th>Framed-Protocol</th>
<th>PPP</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Framed-Protocol</td>
<td>SLIP</td>
<td>2</td>
</tr>
</tbody>
</table>

# Framed Routing Values

<table>
<thead>
<tr>
<th>VALUE</th>
<th>Framed-Routing</th>
<th>None</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Framed-Routing</td>
<td>Broadcast</td>
<td>1</td>
</tr>
</tbody>
</table>
VALUE Framed-Routing Listen 2
VALUE Framed-Routing Broadcast.Listen 3

# Framed Compression Types

VALUE Framed-Compression None 0
VALUE Framed-Compression Van-Jacobson-TCP-IP 1

# Login Services

VALUE Login-Service Telnet 0
VALUE Login-Service Rlogin 1
VALUE Login-Service TCP-Clear 2
VALUE Login-Service PortMaster 3

# Status Types

VALUE Acct-Status-Type Start 1
VALUE Acct-Status-Type Stop 2

# Authentication Types

VALUE Acct-Authentic RADIUS 1
VALUE Acct-Authentic Local 2

# Termination Options

VALUE Termination-Action Default 0
VALUE Termination-Action RADIUS-Request 1

# NAS Port Types, available in ComOS 3.3.1 and later

VALUE NAS-Port-Type Async 0
VALUE NAS-Port-Type Sync 1
VALUE NAS-Port-Type ISDN 2
VALUE NAS-Port-Type ISDN-V120 3
VALUE NAS-Port-Type ISDN-V110 4

# Acct Terminate Causes, available in ComOS 3.3.2 and later

VALUE Acct-Terminate-Cause User-Request 1
VALUE Acct-Terminate-Cause Lost-Carrier 2
### RADIUS Dictionary

<table>
<thead>
<tr>
<th>VALUE</th>
<th>Acct-Terminate-Cause</th>
<th>Lost-Service</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Idle-Timeout</td>
<td>4</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Session-Timeout</td>
<td>5</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Admin-Reset</td>
<td>6</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Admin-Reboot</td>
<td>7</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Port-Error</td>
<td>8</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>NAS-Error</td>
<td>9</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>NAS-Request</td>
<td>10</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>NAS-Reboot</td>
<td>11</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Port-Unneeded</td>
<td>12</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Port-Preempted</td>
<td>13</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Port-Suspended</td>
<td>14</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Service-Unavailable</td>
<td>15</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Callback</td>
<td>16</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>User-Error</td>
<td>17</td>
</tr>
<tr>
<td>VALUE</td>
<td>Acct-Terminate-Cause</td>
<td>Host-Request</td>
<td>18</td>
</tr>
</tbody>
</table>

| VALUE          | Tunnel-Type                             | L2TP           | 3  |
| VALUE          | Tunnel-Medium-Type                      | PPP            | 1  |

# Non-Protocol Integer Translations

| VALUE      | Auth-Type                      | Local      | 0  |
| VALUE      | Auth-Type                      | System     | 1  |
| VALUE      | Auth-Type                      | SecurID     | 2  |
| VALUE      | Auth-Type                      | Crypt-Local| 3  |
| VALUE      | Auth-Type                      | Reject     | 4  |
| VALUE      | Auth-Type                      | ActivCard   | 5  |

# Configuration Values

# comment out these two lines to turn account expiration off

| VALUE      | Server-Config                        | Password-Expiration | 30 |
| VALUE      | Server-Config                        | Password-Warning    | 5  |
This appendix provides contact information for the makers of the following third-party products that might be used with RADIUS for UNIX:

- ActivCard
- iPass
- SecurID

Lucent Remote Access recommends that you visit the corporate website for the product you are interested in to find the most current and complete contact information.

**ActivCard**

ActivCard products are manufactured by ActivCard, Inc.

**Website**

http://www.activcard.com

**Voice**

+33-0-1-42-04-84-00 in Europe  
+65-775-3844 in Singapore  
+1-510-574-0100 in the United States

**Fax**

+33-0-1-42-04-84-84 in Europe  
+65-775-3044 in Singapore  
+1-510-574-0101 in the United States
Email

In Asia-Pacific, contact techsup@activcard.fr.
In Europe, contact techsup@activcard.fr.
In the United States, contact techsup@activcard.com.

iPass services are provided by iPass, Inc.

Website

http://www.ipass.com

Voice

+65-334-8783 in Singapore
+1-650-237-7300 in the United States

Fax

+65-336-6933 in Singapore
+1-650-237-7321 in the United States

Email

In Asia Pacific, contact AP@ipass.com.
In Europe, contact EU@ipass.com.
In Latin America, contact LA@ipass.com.
In North America, contact NA@ipass.com.
In the rest of the world, contact RW@ipass.com.
SecurID

SecurID products are manufactured by Security Dynamics Technologies, Inc.

Website

http://www.securid.com

Voice

+44-118-936-2699 in Europe
+1-781-687-7700 in the United States

Tollfree numbers

Belgium 0800-7-5216
Denmark +800-732-8743-1
Finland +800-732-8743-1
France +800-732-8743-1
Germany +800-732-8743-1
Holland +800-732-8743-1
Italy 1677-90847
Norway +800-732-8743-1
Spain 900-9708918
Sweden +800-732-8743-1
Switzerland +800-732-8743-1
United Kingdom 0800-072-5095
United States 800-995-5095
Fax

+44-118-979-5833 in Europe
+1-781-687-7016 in the United States

Email

Contact info@securid.com.
Index

Symbols
/etc/raddb/users, checking A-2
@ in named realm 9-3

A
access filters 4-34
access-reject packet
    immediate packet A-6
    multiple packets A-6
    result of debug output A-6
access rights, administrative 4-24
accounting
    accepting unsigned records 8-5, 8-10
    attributes 8-3, 8-6
    configuring options on a UNIX host 8-5
    error messages B-1
    example records 8-11
    logged information 8-6
    overview 8-1
    primary and secondary server 2-2, 3-3
    server configuration 8-4
    server requirements 8-3
    server version 8-5
    specifying alternate directory for logs 8-5
    start and stop records 8-6
accounting signatures 1-6
Acct-Authentic 8-6
Acct-Delay-Time 8-6
Acct-Input-Octets 8-6
Acct-Output-Octets 8-6
Acct-Session-Id 8-6
Acct-Session-Time 8-7

Acct-Status-Type 8-7
Acct-Terminate-Cause 8-7
ACE/Server
    administration 7-7
    installation on a UNIX host 7-4
    port numbers 7-6
    requirements for UNIX 7-4
    service names 7-6
    starting and stopping on UNIX 7-7
ActivAdmin 6-2
ActivCard
    ActivAdmin 6-2
    ActivCoupler 6-3
    ActivEngine 6-2
    ActivEngine API 6-3
    ActivEngine installation on a UNIX host 6-4
    asynchronous dynamic passwords 6-2
    components 6-2
    config.aeg file 6-7
    configuring RADIUS for 6-7
    contact information E-1
    definition 6-1
    requirements for UNIX 6-1
    run sradiusd with 6-3
    synchronous dynamic passwords 6-2
    technical support 6-1
    testing with aegtest utility 6-6
    token 6-2
    working with RADIUS 6-3
ActivCoupler 6-3
ActivEngine 6-2
ActivEngine API 6-3
additional references xiii
address binding 1-7
addresses, applying subnet masks to 4-27
administration, ACE/Server 7-7
administrative rights, granting to users 4-24
Administrative-User value for Service-Type reply item 4-22
Admin-Reboot session termination 8-7
Admin-Reset session termination 8-7
Advice-of-Charge accounting attribute 8-9
aegtest ActivEngine utility 6-6
alias for remote server 9-3
applying
  access filters 4-34
  packet filters to session 4-30
asynchronous dynamic passwords 6-2
attributes, accounting
  Acct-Authentic 8-6
  Acct-Delay-Time 8-6
  Acct-Input-Octets 8-6
  Acct-Output-Octets 8-6
  Acct-Session-Id 8-6
  Acct-Session-Time 8-7
  Acct-Status-Type 8-7
  Acct-Terminate-Cause 8-7
  Called-Station-Id 8-9
  Calling-Station-Id 8-9
  LE-Advice-of-Charge 8-9
  LE-Terminate-Detail 8-9
  NAS-Port-Type 8-10
  overview 8-6
  Request-Authenticator 8-10
  Timestamp 8-9
attributes, vendor-specific 1-7, 8-9
authenticating
  users with Call-Check 4-17
authentication
  callback 4-24
  error messages B-3
  overview 1-3
  primary server 2-1, 3-3
  restricting to a group of users 4-16
  secondary server 2-2, 3-3
  specifying type 4-9
  troubleshooting A-1
authentication shell 7-8
authorization, overview 1-4
Auth-Type check item
  Local value 4-9
  Reject value 4-10
  SecurID value 4-10
  System value 4-10

B
binding RADIUS to a specific IP address 1-7
Broadcast-Listen value for Framed-Routing reply item 4-29
Broadcast value for Framed-Routing reply item 4-29
builddbm
  files generated by 4-40
  utility 1-9, 4-40

C
callback, authenticating users with 4-24
Callback-Framed-User value for Service-Type reply item 4-22
Callback-Login-User
  configuring 4-31
Callback-Login-User value for Service-Type reply item 4-22
Callback-Number reply item 4-25
Callback session termination 8-7
call-check, authenticating users with 4-17
Call-Check-User value for Service-Type reply item 4-23
Call-Check value for Service-Type check item 4-17
Called-Station-Id
    accounting attribute  8-9
    check item  4-14
    numbered realm  9-3
Calling-Station-Id
    accounting attribute  8-9
    check item  4-14
caution icon  xv
challenge handshake authentication protocol 4-39
challenge-response
    ActivCard  6-2
    CHAP  4-39
changing from default  C-2
    CHAP  4-12,  4-39
check items
    Auth-Type  4-9
    Called-Station-Id  4-14
    Calling-Station-Id  4-14
    Connect-Rate  4-15
    Crypt-Password  4-11
    definition  4-3
    Expiration  4-11
    Framed-Protocol  4-16
    Group  4-16
    NAS-IP-Address  4-15
    NAS-Port  4-15
    NAS-Port-Type  4-15
    overview  4-6
    Password  4-9
    Prefix  4-12
    Service-Type  4-17
    Suffix  4-12
clearinghouse server
    proxy entry for  9-9
clients
    configuring PortMaster  3-1
    configuring with command line  3-2
    configuring with PMVision  3-4
error messages  B-9
    NAS-IP-Address  4-15
    NAS-Port  4-15
    NAS-Port-Type  4-15
clients file
    definition  1-9
    modifying  3-1
    with proxy service  9-10
compression, TCP/IP  4-26
config.aeg file for ActivCard  6-7
configuration
    IPX network connection  4-29
    login users  4-31
    PPP users  4-25
    proxy service  9-7
    RIP on user’s interface  4-29
    SLIP users  4-25
configuring
    caching on UNIX hosts  4-40
    overview of RADIUS  1-10
    RADIUS client using command line  3-2
    RADIUS client using PMVision  3-4
    RADIUS menus  5-1
    user information  4-1
connection rate, maximum  4-15
    Connect-Rate check item  4-15
contact information
    ActivCard  E-1
    Europe, Middle East, and Africa  xvi
    iPass  E-1
    Lucent Remote Access technical support  xvi
    North America, Latin America, and Asia Pacific  xvi
    SecurID  E-1
    technical support  xv
    users mailing lists  xvii
conventions in this guide  xiv
Crypt-Password check item  4-11
Framed-MTU reply item  4-26
Framed-Protocol  4-16
    check item  4-16
Framed-Protocol reply item  4-25
Framed-Route reply item  4-28
Framed-Routing  4-29
Framed-Routing options  4-29
Framed-Routing reply item
    Broadcast-Listen value  4-29
    Broadcast value  4-29
    Listen value  4-29
    None value  4-29
Framed-User value
    Service-Type check item  4-18
Framed-User value, Service-Type reply item  4-23

G
granting administrative rights to users  4-24
granting outbound Telnet access to user  4-34
Group check item  4-16
GUI mode, sdadmin  7-8

H
host, specifying for user login  4-32
Host-Request session termination  8-7
host unavailable message  A-3

I
Idle-Timeout
    reply items  4-36
    session termination  8-7
in.pmd  4-32
installation
    ACE/Server on a UNIX host  7-4
    on a UNIX host  2-3
    overview of RADIUS installation and configuration  1-10
with pminstall  2-3
invalid accounting records, accepting  8-5,  8-10
invalid login message  A-4
iPass
    contact information  E-1
    instead of RADIUS protocol in proxy service  9-9
    protocol  1-5
    running iradiusd with  1-5
IPX
    converting decimal to dotted decimal  4-30
    setting network information  4-29

L
L2TP and Call-Check  4-17
layer 2 tunneling protocol and Call-Check  4-17
LE-Advice-of-Charge  8-9
LE-Terminate-Detail  8-9
Listen value for Framed-Routing reply item  4-29
Local value for Auth-Type check item  4-9
Login-IP-Host reply item  4-32
logins
    invalid  A-4
    matching user profiles with  4-3
    restricting  1-6
Login-Service reply item  4-32
    PortMaster value  4-32
    Rlogin value  4-32
    TCP-Clear value  4-32
    Telnet value  4-32
Login-TCP-Port reply item  4-33
Login-User, configuring  4-31
login users, configuring  4-31
Login-User value for Service-Type reply item  4-23
Lost-Carrier session termination  8-8
Lost-Service session termination  8-8
Index

M
mailing lists, subscribing to users xvii
maximum connection rate 4-15
Menu reply item 4-37
menus
  DEFAULT 5-1
  error messages B-14
  format 5-1
  nested 5-3
  reference 5-4
  single-level 5-2
  subdirectory 1-9
  termination 5-4
miscellaneous error messages B-14
multihomed hosts and address binding 1-7

N
NAS-Error session termination 8-8
NAS information
  NAS-IP-Address check item 4-15
  NAS-Port check item 4-15
  NAS-Port-Type accounting attribute 8-10
  NAS-Port-Type check item 4-15
NAS-Prompt-User value for Service-Type reply item 4-23
NAS-Reboot session termination 8-8
NAS-Request session termination 8-8
nested menus 5-3
NIS map 7-6
None value for Framed-Routing reply item 4-29
NOREALM proxy entry 9-9
note icon xv

O
operating systems supported 1-2
options
  radiusd C-1
  Outbound-User reply item 4-34
  value for Service-Type check item 4-18
  value for Service-Type reply item 4-23

P
packet filters, applying to session 4-30
PAP 4-12, 4-38
PASSCODE 7-2
password authentication protocol 4-38
Password check item 4-9
passwords
  alternate password file 1-7
  encryption 4-11
  expiration date 4-11
  location of 4-9
Perl script 4-30
PIN
  SecurID assignment 7-10
  system-generated 7-11
  user-created 7-11
pminstall, installing RADIUS with 2-3
PMVision 3-4
Port-Error session termination 8-8
Port-Limit reply item 4-38
port limits 4-38
PortMaster
  checking A-1
  configuring with command line 3-2
  configuring with PMVision 3-4
  Login-Service reply item 4-32
port numbers
  ACE/Server 7-6
  RADIUS 1-10
  specifying for RADIUS 8-5, C-2
port numbers, specifying TCP for a remote host 4-33
Port-Preempted session termination 8-8

Index-6

RADIUS for UNIX Administrator's Guide
Index

ports
   available, controlling number of 4-38
   specifying for user login 4-33
   virtual 1-6
Port-Suspended session termination 8-8
Port-Unneeded session termination 8-8
PPP, example DEFAULT user profile for 4-12
PPP users, configuring 4-25
Prefix check item 4-12
   in DEFAULT user profiles 4-5
proxy file
   DEFAULT realm entry 9-9
   definition 1-9
   NOREALM realm entry 9-9
proxy server shared secret 9-8
proxy service
   clients file 9-10
   communication with PortMaster 9-2
   configuring 9-7
   DEFAULT remote server 9-7
   default UDP ports 9-5
   definition 9-1
   explained 9-2
   forwarding server 9-2
   named realm 9-3
   numbered realm 9-3
   overview 9-1
   proxy file 9-7, 9-8, 9-10
   proxy file on remote server 9-9
   RADIUS requirements 9-5
   realm 9-3
   remote server 9-2, 9-7
   roaming 9-1
   stop proxy entry 9-9
ptrace, using A-6
punctuation in telephone numbers 4-25

R
RADIUS
   accounting 8-1
   actions in UNIX C-1
   authentication 1-3
   authorization 1-4
   client configuration 3-1
   configuring for SecurID 7-10
   dictionary file D-1
   directory structure 1-8
   enhancements for version 2.1 1-5
   features 1-1
   functions 1-3
   installation and configuration overview 1-10
   primary accounting server 2-2
   primary authentication server 2-1
   proxy service 9-1
   secondary accounting server 2-2
   secondary authentication server 2-2
   server requirements 2-1
   users file 4-1
   working with SecurID 7-3
radiusd
   checking A-1
   running with users file in DBM format 4-40
radiusd flags or options
   accounting 8-5
   complete list C-1
RADIUS security
   proxy service 9-1, 9-7
RADPASS 1-8
realm 9-3
   named 9-3
   numbered 9-3
realms
   DEFAULT 9-9
   NOREALM 9-9
   special named realms 9-9
Index

references
  books  xiv
  RFCs  xiii
Reject value for Auth-Type check item  4-10
related documentation  xi
remote server for proxy service
  configuring  9-10
remote servers
  DEFAULT for proxy service  9-7
  for proxy service, how they work  9-2,  9-6
  See also proxy service
reply items
  Callback-Login-User  4-31
  Callback-Number  4-25
  definition  4-3
  examples  4-40
  Framed-Compression, Van Jacobson header compression  4-26
  Framed-IP-Netmask  4-27
  Framed-IPX-Network  4-29
  Framed-MTU  4-26
  Framed-Protocol  4-25
  Framed-Route  4-28
  Framed-Routing  4-29
  Idle-Timeout  4-36
  Login-IP-Host  4-32
  Login-Service value  4-32
  Login-TCP-Port  4-33
  Login-User  4-31
  Menu  4-37
  Outbound-User  4-34
  overview  4-19
  Port-Limit  4-38
  Session-Timeout  4-36
  Termination-Menu  4-37
  timeouts  4-35
Request-Authenticator  8-10
restricting logins  1-6
rights, granting administrative to users  4-24
RIP
  configuring on user’s interface  4-29
  options  4-29
Rlogin value for Login-Service reply item  4-32
roaming  9-1
  defined  9-1
  with iPass  1-5
routing table, adding a route to  4-28
S
  sdadmin utility  7-4,  7-7,  7-8
  sdconnect utility  7-7
  sdsetup utility  7-6
  sdshell utility  7-8
  secret
  shared, proxy server and client  9-8
  secret, shared  2-2,  3-3
SecurID
  ACE/Server installation on a UNIX host  7-4
  ACE/Server requirements for UNIX  7-4
  authentication shell  7-8
  configuring RADIUS for  7-10
  contact information  E-1
  error messages  B-16
  new users  7-10
  PIN assignment  7-10
  port numbers  7-6
  run sradiusd with  7-3
  sdadmin  7-7
  service names  7-6
  technical support  7-1,  7-9
  troubleshooting  7-13
  value for Auth-Type check item  4-10
  working with RADIUS  7-3
server
  alias for remote  9-3
  forwarding  9-2
  remote  9-2
Index

server, RADIUS
  accounting requirements 8-3
  primary accounting 2-2
  primary authentication 2-1
  requirements 2-1
  secondary accounting 2-2
  secondary authentication 2-2

services
  types of 4-22

Service-Type check item
  Call-Check value 4-17
  Framed-User value 4-18
  Outbound-User value 4-18

Service-Type reply item
  Administrative-User value 4-22, 4-24
  Callback-Framed-User value 4-22, 4-25
  Callback-Login-User value 4-22, 4-25, 4-31
  Call-Check-User value 4-23
  Framed-User value 4-23, 4-25
  Login-User value 4-23, 4-31
  NAS-Prompt-User value 4-23, 4-24
  Outbound-User value 4-23

Service-Types 4-22

Service Unavailable session termination 8-8

session termination, reasons for
  Admin-Reboot 8-7
  Admin-Reset 8-7
  Callback 8-7
  Host-Request 8-7
  Idle-Timeout 8-7
  Lost-Carrier 8-8
  Lost-Service 8-8
  NAS-Error 8-8
  NAS-Reboot 8-8
  NAS-Request 8-8
  Port-Error 8-8
  Port-Preempted 8-8
  Port-Suspended 8-8
  Port-Unneeded 8-8
  Service Unavailable 8-8
  Session-Timeout 8-8
  User-Error 8-9
  User-Request 8-9

Session-Timeout
  reply items 4-36
  session termination 8-8

shared secret 2-2, 3-3
  proxy server and client 9-8

SIGUSR1 and SIGUSR2, using 1-8, A-5

single-level menus 5-2

SLIP, example DEFAULT user profile for 4-12

SLIP users, configuring 4-25

specifying
  host for user login 4-32
  port for user login 4-33

sradius 6-3

subnet masks 4-27

Suffix check item
  in DEFAULT user profile 4-5
  overview 4-12

support, technical xv, xvi
  ActivCard 6-1
  Security Dynamics 7-1, 7-9

synchronous dynamic passwords 6-2

syslog
  improvements 1-8
  sending debug output to C-2
  troubleshooting with 1-8

system-generated PIN 7-11

System value for Auth-Type check item 4-10

T

tables
  routing 4-28

TCP/IP header compression 4-26

TCP-clear tunneling and Call-Check 4-17

TCP-Clear value for Login-Service reply item 4-32
technical support  xv
  ActivCard  6-1
  Lucent Remote Access  xvi
  Security Dynamics  7-1,  7-9
 telephone numbers, punctuation in  4-25
 Telnet
  example DEFAULT user profile for  4-12
  granting user outbound access  4-34
  Login-Service reply item  4-32
 Terminate-Detail accounting attribute  8-9
 terminating proxy  9-9
 termination causes. See session termination, reasons for
 Termination-Menu reply items  4-37
 Termination menus  5-4
 text files
  clients  3-1
  menu  5-1
  users  4-1
 timeouts
  setting idle  4-36
  setting session  4-36
  setting user  4-35
 Timestamp  8-9
 token  6-2
  definition  7-2
tokencode
  definition  7-2
  invalid  7-12
 troubleshooting
  ActivCard  6-9
  authentication  A-1
  SecurID  7-13
  turning on debugging  1-8,  A-5,  C-2
  with the dictionary  A-2
types of service  4-22

U
UDP
  default ports  9-5
UNIX commands  C-1
unsigned accounting records, accepting  8-5,
  8-10
user-created PIN  7-11
User-Error session termination  8-9
username
  definition  4-2
  prefix and suffix to  4-12
user profile
  components  4-2
  DEFAULT  4-4
  editing  4-4
  example  4-40
  format  4-2
  overview  4-1
user profiles
  matching logins with  4-3
User-Request session termination  8-9
users
  Callback-Framed-User  4-25
  Callback-Login-User  4-25
  Call-Check  4-17
  disconnecting  4-35
  error messages  B-17
  Framed-User  4-25
  granting outbound Telnet access  4-34
  groups of  4-16
  login  4-31
  NAS-Prompt-User  4-24
  port limits  4-38
  PPP, configuring  4-25
  SLIP, configuring  4-25
users file  1-9,  4-1
  checking  A-2
  converting from text to DBM  4-40
Index

utilities
   aegtest  6-6
   builddbm  1-9,  4-40
   pminstall  2-3
   sdadmin  7-4,  7-8
   sdconnect  7-7
   sdsetup  7-6
   sdshell  7-8

V
vendor-specific attributes  1-7,  8-9
version
   determining for accounting server  8-5
Virtual  1-6
vports file  1-6
definition  1-9

W
warning icon  xv