Juniper Networks
Steel-Belted Radius

Administration Guide
Enterprise Edition

Release 6.0
February 2007
# Table of Contents

## About This Guide

- Before You Begin ............................................................................................ xi
- Audience ......................................................................................................... xi
- What’s In This Manual .................................................................................... xi
- Typographical Conventions
  - Editions/Used In ..................................................................................... xiv
  - Syntax .................................................................................................... xiv
- Related Documentation
  - Steel-Belted Radius Documentation ....................................................... xv
  - Requests for Comments (RFCs) ............................................................... xv
  - Third-Party Products ................................................................................. xvi
- Contacting Technical Support ........................................................................ xvi

## Chapter 1

### About Steel-Belted Radius

- Steel-Belted Radius Features ........................................................................ 1
- Licensing ........................................................................................................ 2

## Chapter 2

### RADIUS Basics

- RADIUS Overview ........................................................................................ 3
- RADIUS Packets .......................................................................................... 5
- RADIUS Ports .............................................................................................. 6
- RADIUS Configuration ................................................................................ 6
- Multiple RADIUS Servers .......................................................................... 7
- Shared Secrets .............................................................................................. 8
- Accounting ................................................................................................... 10
- Attributes ..................................................................................................... 10
- Dictionaries .................................................................................................. 10
- User Attribute Lists ..................................................................................... 11
- Attribute Values .......................................................................................... 12
- Default Values ............................................................................................ 13
- Centralized Configuration Management ................................................... 14
- Proxy RADIUS
  - Proxy RADIUS Authentication ............................................................... 15
  - Proxy RADIUS Accounting .................................................................... 16
- Authentication .............................................................................................. 16
  - Authentication Methods ........................................................................... 17
  - Configuring the Authentication Sequence ............................................. 19
  - Configuring Authentication Methods ..................................................... 19
  - Two-Factor Authentication ...................................................................... 21
- Password Protocols ...................................................................................... 22
  - Password Authentication Protocol .......................................................... 22
  - Challenge Handshake Authentication Protocol .................................... 23
  - MS-CHAP and MS-CHAP-V2 ................................................................. 23
Steel-Belted Radius Administration Guide

Table of Contents

Accounting .....................................................................................................24
   Accounting Sequence ...............................................................................24
   RADIUS Client Groups ...........................................................................26
IP Address Assignment .............................................................................27
   Address Pools and Replication ...............................................................27
   Hints .......................................................................................................28
Resource Management ...............................................................................29
   Network Address Assignment ................................................................29
   Concurrent Network Connections ..........................................................31
   Phantom Records ..................................................................................32

Chapter 3 Using SBR Administrator 33

   Running the SBR Administrator ..............................................................33
   Navigating in SBR Administrator ............................................................34
      SBR Administrator Panels ................................................................34
      SBR Administrator Sidebar ..................................................................35
      SBR Administrator Menus .....................................................................35
      SBR Administrator Toolbar ..................................................................37
      SBR Administrator Dialogs .................................................................37
   Adding License Keys ...............................................................................39
   Accessing Online Help .............................................................................40
   Displaying Version Information .............................................................40
   Exiting the SBR Administrator ..............................................................40

Chapter 4 Administering RADIUS Clients 41

   RADIUS Clients Panel .............................................................................41
   Adding a RADIUS Client or Client Group ..................................................42
   Verifying a Shared Secret ..........................................................................45
   Deleting a RADIUS Client .......................................................................46

Chapter 5 Administering Users 47

   User Files ..................................................................................................47
   Users Panels .............................................................................................47
   Setting Up Native Users ...........................................................................48
      Adding a Native User ............................................................................48
      Editing a Native User ...........................................................................50
      Deleting a Native User .........................................................................51
      Adding a Checklist or Return List Attribute for a User .......................52
   Setting Up Windows Domain Users ...........................................................53
      Prequalification Checklists ...................................................................53
      MS-CHAP Considerations .....................................................................54
      Expired Domain Passwords .................................................................55
      Windows Domain Authentication Configuration ..................................55
      Adding a Domain User or Domain Group ............................................55
   Setting Up SecurID Users .........................................................................58
   Setting Up TACACS+ Users .....................................................................60
   Setting Up UNIX Users ............................................................................62
   Editing User Settings ..............................................................................64
      Selecting a Profile ..................................................................................64
      Setting Attribute Values ........................................................................64
      Removing Attribute/Value Pairs ............................................................65
      Reordering Attributes ............................................................................65
### Table of Contents

**Chapter 6 Administering Profiles**  
- About Profiles .......................... 67
- Adding a Checklist or Return List Attribute for a Profile .......................... 67
- Resolving Profile and User Attributes ..................................................... 67
- Setting Up Profiles .................................................................................. 68
- Adding a Profile ....................................................................................... 68
- Removing a Profile .................................................................................. 70

**Chapter 7 Administering Proxy RADIUS**  
- About Proxy RADIUS ............................................................................. 71
- Proxy RADIUS Authentication ................................................................ 71
- Proxy RADIUS Accounting .................................................................... 72
- Adding a Proxy Target ............................................................................ 72
- Maintaining an Accounting Shared Secret ............................................. 74
- Deleting a Proxy Target ......................................................................... 75
- Steel-Belted Radius as a Target ............................................................. 75
  - Dictionaries when Steel-Belted Radius is the Target .......................... 75
  - Accepting Packets from Any Proxy ................................................ 76
  - Proxy RADIUS as an Authentication Method ....................................... 76

**Chapter 8 Administering RADIUS Tunnels**  
- About RADIUS Tunnels ........................................................................ 77
- Tunnel Authentication Sequence ............................................................. 78
- Configuring Tunnel Support ................................................................. 79
- Concurrent Tunnel Connections ............................................................ 79
- Configuring RADIUS Tunnels .............................................................. 80
- Adding a Tunnel ...................................................................................... 80
- Editing a Tunnel ...................................................................................... 83
- Deleting a Tunnel .................................................................................... 83
- Configuring Tunnel Name Parsing ......................................................... 84

**Chapter 9 Administering Address Pools**  
- Address Pool Files ................................................................................. 85
- Setting Up IP Address Pools ................................................................. 85
- Adding an IPv4 Address Pool ................................................................ 86
- Editing an IP Address Pool ................................................................... 87
- Removing an IP Address Pool ............................................................... 88
- Specifying an IP Address Pool for User/Profile Records ...................... 88
- Setting Up IPX Address Pools .............................................................. 88
- Adding an IPX Pool ............................................................................... 89
- Editing an IPX Address Pool ............................................................... 90
- Removing an IPX Address Pool ........................................................... 91
- Specifying Pooled IPX Network Numbers in User/Profile Records ........ 91
<table>
<thead>
<tr>
<th>Chapter 14</th>
<th>LDAP Configuration Interface</th>
<th>165</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Configuration Interface File</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>About the LDAP Configuration Interface</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>LDAP Utilities</td>
<td>166</td>
<td></td>
</tr>
<tr>
<td>LDAP Requests</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>Downloading the LDAP Utilities (Windows)</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>LDAP Version Compliance</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>Configuring the LDAP TCP Port</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>Configuring the LCI Password</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>LDAP Virtual Schema</td>
<td>169</td>
<td></td>
</tr>
<tr>
<td>LDAP Rules and Limitations</td>
<td>174</td>
<td></td>
</tr>
<tr>
<td>LDAP Command Examples</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>Searching for Records</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>Modifying Records</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>Importing Records From Another LDAP Database</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>Deleting Records</td>
<td>181</td>
<td></td>
</tr>
<tr>
<td>LDIF File Examples</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>Adding RADIUS Clients with LDIF</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>Adding Users with LDIF</td>
<td>183</td>
<td></td>
</tr>
<tr>
<td>Adding Proxy Targets with LDIF</td>
<td>185</td>
<td></td>
</tr>
<tr>
<td>Adding Tunnels with LDIF</td>
<td>186</td>
<td></td>
</tr>
<tr>
<td>Adding IP Address Pools with LDIF</td>
<td>187</td>
<td></td>
</tr>
<tr>
<td>Adding IPX Address Pools with LDIF</td>
<td>187</td>
<td></td>
</tr>
<tr>
<td>Configuring a RADIUS Server with LDIF</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td>Statistics Variables</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>Counter Statistics</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>Rate Statistics</td>
<td>190</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15</th>
<th>Configuring SQL Authentication</th>
<th>193</th>
</tr>
</thead>
<tbody>
<tr>
<td>About SQL Authentication</td>
<td>193</td>
<td></td>
</tr>
<tr>
<td>SQL Authentication Process</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>Stored Procedures</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>Connectivity Issues</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>Configuring SQL Authentication</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>Files</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>Using the SQL Authentication Header File</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>Using Multiple SQL Authentication Methods</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>Connecting to the SQL Database</td>
<td>197</td>
<td></td>
</tr>
<tr>
<td>SQL Statement Construction</td>
<td>198</td>
<td></td>
</tr>
<tr>
<td>Password Parameters</td>
<td>199</td>
<td></td>
</tr>
<tr>
<td>Overlapped Execution of SQL Statements</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>%result Parameter</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>SQL Authentication and Password Format</td>
<td>201</td>
<td></td>
</tr>
<tr>
<td>Working With Stored Procedures in Oracle</td>
<td>202</td>
<td></td>
</tr>
<tr>
<td>Working With Stored Procedures in MS-SQL</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>Example 1</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>Example 2</td>
<td>204</td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

## Chapter 16 Configuring SQL Accounting  207

- About SQL Accounting ............................................................... 207
- Stored Procedures ........................................................................ 208
- Connectivity Issues ..................................................................... 208
- Configuring SQL Accounting ...................................................... 209
  - Files ......................................................................................... 209
  - Using the SQL Accounting Header File .................................... 210
  - Using Multiple SQL Databases ................................................. 210
- Connecting to the SQL Database .................................................. 210
- SQL Statement Construction .......................................................... 211
  - INSERT Statement and VALUES Section ................................. 212
  - Using Multiple SQL Statements .............................................. 215
- Overlapped Execution of SQL Statements ....................................... 215
- SQL Accounting Return Values .................................................... 215
- Accounting Stored Procedure Example ........................................ 216

## Chapter 17 Configuring LDAP Authentication  219

- About LDAP Authentication ......................................................... 219
- LDAP Variable Table .................................................................... 220
- Types of LDAP Authentication ...................................................... 220
- Configuring LDAP Authentication .................................................. 222
  - Files ......................................................................................... 223
  - LDAP Database Schema .......................................................... 223
  - LDAP Authentication and Password Format ................................ 225
- LDAP Authentication Sequence ...................................................... 226
- LDAP Authentication Examples ..................................................... 227
  - Bind Authentication with Default Profile .................................... 227
  - BindName Authentication with Callback Number Returned .......... 228
  - LDAP Bind with Profile Based on Network Access Device ............ 229

## Chapter 18 Displaying Statistics  231

- Displaying Authentication Statistics .............................................. 231
- Displaying Accounting Statistics .................................................... 232
- Displaying Proxyed Request Statistics ............................................ 234
- Displaying RADIUS Client Statistics ............................................. 235
- Displaying RADIUS Proxy Targets Statistics ................................. 237
- Displaying IP Address Pool Statistics ............................................ 238

## Chapter 19 Logging and Reporting  239

- Logging Files ................................................................................. 239
- Displaying the Current Sessions List .............................................. 239
  - Searching the Current Sessions List ............................................ 241
  - Deleting Entries from the Sessions List ....................................... 242
- Displaying the Authentication Log Files .......................................... 242
  - File Permissions for Log Files (Solaris/Linux) .............................. 242
  - Enabling and Disabling the Authentication Log Files .................... 247
  - Viewing the Authentication Log Files ......................................... 247
  - Saving the Log Files ................................................................. 248
  - Searching the Log Files ............................................................. 249
- Configuring the Log Retention Period ............................................ 250
- Using the Server Log File .............................................................. 251
  - Level of Logging Detail .............................................................. 252
Using the Accounting Log File ................................................................. 252
  Accounting Log File Format ................................................................. 252
  First Line Headings ............................................................................ 253
  Comma Placeholders ........................................................................... 254
  Standard RADIUS Accounting Attributes ........................................... 254

Appendix A  Glossary .................................................................................. 257

Appendix B  When to Restart Steel-Belted Radius ....................................... 263

Appendix C  Technical Notes ........................................................................ 265
  LDAP Support for Novell eDirectory ....................................................... 265
  CCA Support for 3COM ........................................................................ 269
    Configuration ........................................................................................ 269
    Setting User and Profile Attributes ...................................................... 269
  Ascend Filter Translation ......................................................................... 270
    Configuration ........................................................................................ 270
    Syntax .................................................................................................. 270
  Ericsson Enhanced Token Caching .......................................................... 271
    Enhanced Token Caching Configuration ................................................. 272
    Enhanced Token Caching Administration .............................................. 273
  Uniport Plug-In ..................................................................................... 273
    Operation ............................................................................................ 273
    Configuration ........................................................................................ 274
  Windows Performance Monitor ............................................................... 275

Appendix D  Authentication Protocols ........................................................ 281

Appendix E  Importing and Exporting Data .................................................. 283
  Exporting to a RADIUS Information File ................................................ 283
    Importing Into the Steel-Belted Radius Database ................................... 285

Appendix F  Configuring IPv6 Support ......................................................... 287
  About IPv6 ............................................................................................ 287
    IPv6 and Steel-Belted Radius ............................................................... 287
    IPv6 Features ....................................................................................... 288
    IPv6 Addressing .................................................................................. 288
    IPv6 Support in Steel-Belted Radius .................................................... 296
    RADIUS IPv6 Attributes .................................................................... 299
  Configuring IPv6 ................................................................................... 303
    Enabling IPv6 Networking .................................................................. 303
    Configuring IPv6 Scope IDs ................................................................. 303
    Configuring IPv6 Addresses for RADIUS Client Connections ............... 304
    Configuring DNSv6 Support ............................................................... 304

Appendix G  Stopping and Starting Steel-Belted Radius ............................... 305
  Stopping the Steel-Belted Radius Server ................................................ 305
    Windows ............................................................................................. 305
    Solaris ................................................................................................. 306
    Linux ................................................................................................... 306
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting the Steel-Belted Radius Server</td>
<td>307</td>
</tr>
<tr>
<td>Windows</td>
<td>307</td>
</tr>
<tr>
<td>Solaris</td>
<td>307</td>
</tr>
<tr>
<td>Linux</td>
<td>307</td>
</tr>
<tr>
<td>Displaying RADIUS Status Information (Linux)</td>
<td>308</td>
</tr>
</tbody>
</table>

**Index**

309
About This Guide

The Steel-Belted Radius Administration Guide/Enterprise Edition describes how to configure and administer the Steel-Belted Radius software on a server running the Solaris operating system, the Linux operating system, or the Windows XP/Windows Server 2003 operating system.

Before You Begin

This manual assumes that you have installed the Steel-Belted Radius software and the SBR Administrator. For more information, refer to the Steel-Belted Radius Getting Started manual.

Audience

This manual is intended for network administrators responsible for implementing and maintaining authentication, authorization, and accounting services for an enterprise. This manual assumes that you are familiar with general RADIUS and networking concepts and the specific environment in which you are installing Steel-Belted Radius.

If you use Steel-Belted Radius with third-party products such as Oracle or RSA SecurID, you should be familiar with their installation, configuration, and use.

What’s In This Manual

This manual contains the following chapters and appendixes:

- Chapter 1, “About Steel-Belted Radius,” presents an overview of Steel-Belted Radius and describes licensing requirements for Steel-Belted Radius.

- Chapter 2, “RADIUS Basics,” summarizes important concepts relating to the operation of Steel-Belted Radius.

- Chapter 3, “Using SBR Administrator,” describes how to use the SBR Administrator to configure Steel-Belted Radius.
Chapter 4, “Administering RADIUS Clients,” describes how to set up a network access device (NAD) as a Steel-Belted Radius client.

Chapter 5, “Administering Users,” describes how to set up users in the Steel-Belted Radius database.

Chapter 6, “Administering Profiles,” describes how to set up user profiles to simplify user administration.

Chapter 7, “Administering Proxy RADIUS,” describes how to identify proxy RADIUS targets.

Chapter 8, “Administering RADIUS Tunnels,” describes how to set up secure RADIUS tunnels.

Chapter 9, “Administering Address Pools,” describes how to set up IPv4 and IPX address pools.

Chapter 10, “Setting Up Administrator Accounts,” describes how to identify who can administer Steel-Belted Radius.

Chapter 11, “Setting Up Filters,” describes how to configure and maintain attribute filters in Steel-Belted Radius.


Chapter 13, “Configuring Replication,” describes how to configure and use the centralized configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment.

Chapter 14, “LDAP Configuration Interface,” describes how to use public domain LDAP utilities to populate a Steel-Belted Radius server database.

Chapter 15, “Configuring SQL Authentication,” describes how to configure authentication against records stored in an external SQL database.

Chapter 16, “Configuring SQL Accounting,” describes how to configure Steel-Belted Radius to write accounting information to an external SQL database.

Chapter 17, “Configuring LDAP Authentication,” describes how to configure authentication against records stored in an external LDAP database.

Chapter 18, “Displaying Statistics,” describes how to use the monitoring facilities in Steel-Belted Radius.

Chapter 19, “Logging and Reporting,” describes how to use the logging and reporting facilities in Steel-Belted Radius.

Appendix A, “Glossary,” provides brief explanations for RADIUS terminology used in this and other Steel-Belted Radius manuals.

Appendix C, “Technical Notes,” presents tips for configuring Steel-Belted Radius to interoperate with equipment and facilities from other vendors.

Appendix D, “Authentication Protocols,” provides a matrix of authentication methods and their supported authentication protocols.

Appendix E, “Importing and Exporting Data,” describes how to import and export information in a Steel-Belted Radius database to and from an XML file.


Appendix G, “Stopping and Starting Steel-Belted Radius,” describes how to stop and restart the Steel-Belted Radius service (Windows) or RADIUS daemon (Solaris/Linux).

**Typographical Conventions**

Table 1 describes the text conventions used throughout this manual.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold typeface</strong></td>
<td>Indicates buttons, field names, dialog names, and other user interface elements.</td>
<td>Use the <strong>Scheduling</strong> and <strong>Appointment</strong> tabs to schedule a meeting.</td>
</tr>
<tr>
<td><strong>Plain sans serif typeface</strong></td>
<td>Represents: Code, commands, and keywords, URLs, file names, and directories.</td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>Code: <code>certAttr.OU = 'Retail Products Group'</code></td>
<td>Code: <code>certAttr.OU = 'Retail Products Group'</code></td>
</tr>
<tr>
<td></td>
<td>URL: Download the JRE application from: <a href="http://java.sun.com/j2se/">http://java.sun.com/j2se/</a></td>
<td>URL: Download the JRE application from: [<a href="http://java.sun.com/j2se/">http://java.sun.com/j2se/</a>]</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>Identifies: Terms defined in text, variable elements, and book names.</td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>Defined term: An <strong>RDP client</strong> is a Windows component that enables a connection between a Windows server and a user’s machine.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Variable element: Use settings in the <strong>Users &gt; Roles &gt; Select Role &gt; Terminal Services</strong> page to create a terminal emulation session.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Book name: See the Steel-Belted Radius <strong>Administration Guide</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
Editions/Used In

Steel-Belted Radius is available in multiple editions to meet the requirements of different types of customers. This manual uses the following abbreviations to identify editions of Steel-Belted Radius:

- GEE – Global Enterprise Edition
- SPE – Service Provider Edition
- SPE + EAP – Service Provider Edition with optional EAP Extension Module
- SPE + MIM – Service Provider Edition with optional Mobile IP Module
- EE – Enterprise Edition

Syntax

- `radiusdir` represents the directory into which Steel-Belted Radius has been installed. By default, this is `C:\Program Files\Juniper Networks\Steel-Belted Radius\Service` for Windows systems and `/opt/juniper/radius` on Linux and Solaris systems.

- Brackets `[ ]` enclose optional items in format and syntax descriptions. In the following example, the first `Attribute` argument is required; you can include an optional second `Attribute` argument by entering a comma and the second argument (but not the square brackets) on the same line.

  \[ <add | replace> = Attribute [Attribute] \]

  In configuration files, brackets identify section headers:

  the [Processing] section of `proxy.ini`

  In screen prompts, brackets indicate the default value. For example, if you press Enter without entering anything at the following prompt, the system uses the indicated default value (`/opt`).

  \[ Enter install path [/opt]: \]

- Angle brackets `< >` enclose a list from which you must choose an item in format and syntax descriptions.

- A vertical bar (`|`) separates items in a list of choices. In the following example, you must specify `add` or `replace` (but not both):

  \[ <add | replace> = Attribute [Attribute] \]
Related Documentation

The following documents supplement the information in this manual.

**Steel-Belted Radius Documentation**

Please review the ReleaseNotes.txt file that accompanies your Steel-Belted Radius software. This file contains the latest information about features, changes, known problems, and resolved problems. If the information the ReleaseNotes.txt file differs from the information found in the Steel-Belted Radius manuals, use the information in the ReleaseNotes.txt file.

In addition to this manual, the Steel-Belted Radius documentation includes the following manuals:

- The Steel-Belted Radius Getting Started Guide describes how to install the Steel-Belted Radius software on a server running the Solaris operating system, the Linux operating system, or the Windows XP/Windows Server 2003 operating system.
- The Steel-Belted Radius Reference Guide describes the configuration options for the Steel-Belted Radius software.
- (SPE Only) The Mobile IP Module Guide describes how the Mobile IP module works and how to configure Steel-Belted Radius/Service Provider Edition to support 3GPP2 or 3GPP services in the Mobile IP module.

**Requests for Comments (RFCs)**

The Internet Engineering Task Force (IETF) maintains an online repository of Request for Comments (RFCs) online at [http://www.ietf.org/rfc.html](http://www.ietf.org/rfc.html). Table 2 lists the RFCs that apply to this guide.

<table>
<thead>
<tr>
<th>RFC Number</th>
<th>Title</th>
</tr>
</thead>
</table>
Third-Party Products

For more information about configuring your access servers and firewalls, consult the manufacturer’s documentation provided with each device.

Contacting Technical Support

For technical support, contact Juniper Networks at support@juniper.net, or at 1-888-314-JTAC (in the United States) or 408-745-9500 (outside the United States).

Check our website (http://www.juniper.net) for additional information and technical notes. When you are running SBR Administrator, you can choose Web > Steel-Belted Radius User Page to access a special home page for Steel-Belted Radius users.

When you call technical support, please have the following at hand:

- Your Steel-Belted Radius product edition and release number (for example, Global Enterprise Edition version 5.4).
- Information about the server configuration and operating system, including any OS patches that have been applied.
- For licensed products under a current maintenance agreement, your license or support contract number.
Question or description of the problem, with as much detail as possible.

Any documentation that may help in resolving the problem, such as error messages, memory dumps, compiler listings, and error logs.
Chapter 1
About Steel-Belted Radius

Thank you for selecting Steel-Belted Radius®/Enterprise Edition.

Steel-Belted Radius is a complete implementation of the RADIUS (Remote Authentication Dial In User Service) protocol. Steel-Belted Radius interfaces with a wide variety of network access equipment, and authenticates remote and wireless LAN (WLAN) users against numerous back-end databases, allowing you to consolidate the administration of all your remote and WLAN users.

Steel-Belted Radius/Enterprise Edition delivers a total RADIUS solution, designed to meet the access control and policy management requirements of enterprises. It interfaces with a wide variety of remote access servers, including VPN and dial-in servers and wireless LAN access points, and authenticates remote and WLAN users against your existing security infrastructure. This lets you control who can access your network and what resources are available to them. Steel-Belted Radius logs all access usage, so you can track and document usage statistics.

Steel-Belted Radius Features

- Centralized management of user access control and security simplifies access administration.
- Flexible, powerful proxy RADIUS features let you easily distribute authentication and accounting requests to the appropriate RADIUS server for processing.
- Authentication against a local database permits network access by employees.
- Flexible authentication options let you use your existing OS-based authentication database, token systems from RSA Security and other vendors, and external SQL/LDAP databases for remote and WLAN user authentication.
- Support for a wide variety of 802.1X-compliant access points and other network access servers ensures compatibility in your network environment.
- You can configure Steel-Belted Radius by means of a graphical SBR Administrator.
Administrative access levels can be defined and applied to user or group accounts on the server machine. Read, write, and read/write access can be applied selectively to various categories of configuration data, including users, RADIUS clients, proxy targets, and statistics.

## Licensing

If you want to install the Steel-Belted Radius server software for a 30-day evaluation, you do not need a license key.

If you want to install a permanent (non-evaluation) copy of Steel-Belted Radius, you must have a single-seat software license key.

If you have more than one copy of the Steel-Belted Radius software installed, you must have a site license key or you must have a separate license key for each installation.

The SBR Administrator can be installed on as many workstations as you require. The SBR Administrator does not require a license key.

For details about licensing, please refer to the Steel-Belted Radius license agreement or contact Juniper Networks.
Chapter 2

**RADIUS Basics**

This chapter presents a conceptual overview of RADIUS (Remote Authentication Dial In User Service) authentication, authorization, and accounting services.

---

**RADIUS Overview**

RADIUS is an industry-standard protocol for providing authentication, authorization, and accounting services.

- **Authentication** is the process of verifying a user’s identity and associating additional information (attributes) to the user’s login session.

- **Authorization** is the process of determining whether the user is allowed on the network and controlling network access values based on a defined security policy.

- **Accounting** is the process of generating log files that record session statistics used for billing, system diagnosis, and usage planning.

A RADIUS-based remote access environment typically involves four types of components:

- An **access client** is a user who initiates a network connection. An access client might be a user dialing into a service provider network, a router at a small office/home office connecting to an enterprise network to provide network access, or a wireless client connecting to an 802.1X access point.

- A **network access device** (NAD), also called a RADIUS client, is a device that recognizes and processes connection requests from outside the network edge. A NAD can be a wireless access point, a modem pool, a network firewall, or any other device that needs to authenticate users. When the NAD receives a user’s connection request, it may perform an initial access negotiation with the user to obtain identity/password information. The NAD then passes this information to the RADIUS server as part of an authentication/authorization request.

---

**NOTE:** The terms “network access device” (NAD), “remote access server” (RAS), and “network access server” (NAS) are interchangeable. This manual uses “NAD,” though some attribute names and parameters retain the older “NAS” in their names.
The RADIUS server matches data from the authentication/authorization request with information in a trusted database, such as the database on the Steel-Belted Radius server or a backend database server. If a match is found and the user’s credentials are correct, the RADIUS server sends an Access-Accept message to the NAD; if a match is not found or a problem is found with the user’s credentials, the server returns an Access-Reject message. The NAD then establishes or terminates the user’s connection. The NAD may then forward accounting information to the RADIUS server to document the transaction; the RADIUS server may store or forward this information as needed to support billing for the services provided.

In some networks, a backend authentication server, such as RSA SecurID or TACACS+; a SQL or LDAP database; or some other RADIUS server for which this server is a proxy, stores the information against which the authentication request is compared. In some cases, the backend server passes information to the RADIUS server, which determines whether a match exists. In other cases, the matching is performed on the backend server, which then passes an accept/reject result to the RADIUS server.

Figure 1 illustrates a simple RADIUS environment.
**RADIUS Packets**

A RADIUS client and RADIUS server communicate by means of RADIUS packets. RADIUS packets carry messages between the RADIUS client and RADIUS server in a series of request/response transactions: the client sends a request and expects a response from the server. If the response does not arrive, the client can retry the request periodically.

Each RADIUS packet supports a specific purpose: authentication or accounting. A packet can contain values called *attributes*. The specific attributes to be found in each packet depend upon the type of packet (authentication or accounting) and the device that sent it (for example, the specific make and model of the NAD device acting as a RADIUS client).

For information on RADIUS authentication packet structures and attributes, see RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*. For information on RADIUS accounting packet structures and attributes, see RFC 2866, *RADIUS Accounting*.

Figure 2 illustrates a simple RADIUS authentication/authorization sequence.

**Figure 2: RADIUS Authentication**

1. The RADIUS access client sends an authentication request containing identification and connection information to the network access device (RADIUS client).

2. When the NAD receives a user’s connection request, it typically performs an initial access negotiation with the user to establish connection information (username, password, network access device identifier, NAD port number, and so on). The NAD then forwards the user information in an authentication request to the RADIUS server.

3. The RADIUS server looks up the user information in a local or remote RADIUS authentication database. The RADIUS server verifies that the user’s name and password are valid. It can also enforce fine-grained security rules by using an *access checklist* to verify specific attributes in the authentication request.
4. If a match is found, the RADIUS server returns an Access-Accept message (4a). The RADIUS server might also send return list information stored in the database, such as the user's authorization or connection parameters, back to the NAD.

If a match is not found, the RADIUS server returns an Access-Reject message (4b).

If third-party software such as RSA SecurID is used, the RADIUS server may prompt the user for more information before accepting or rejecting the authentication request.

5. Based on the information it receives from the RADIUS server, the NAD accepts or refuses the connection request.

After the user is authenticated and the connection established, the NAD may forward accounting data to the RADIUS server to document the transaction; the RADIUS server can store or forward this data to support billing for the services provided.

RADIUS Ports

The RADIUS standard initially used UDP ports 1645 and 1646 for RADIUS authentication and accounting packets. The RADIUS standards group later changed the port assignments to 1812 and 1813, but many organizations still use the old 1645/1646 port numbers for RADIUS.

Any two devices that exchange RADIUS packets must use compatible UDP port numbers. That is, if you are configuring a NAD to exchange authentication packets with a RADIUS server, you must find out which port the server uses to receive authentication packets from its clients (1812, for example). You must then configure the NAD to send authentication packets on the same port (1812). The same is true for RADIUS accounting.

Steel-Belted Radius can listen on multiple ports. For compatibility, the server listens to the old and new default RADIUS ports: ports 1645 and 1812 for authentication, and ports 1646 and 1813 for accounting. To add, change, or disable the ports on which Steel-Belted Radius listens, modify the radius.ini file or edit the services (Windows) or /etc/services (Solaris/Linux) file. The radius.ini file and the services file are described in the Steel-Belted Radius Reference Guide.

RADIUS Configuration

You must configure a RADIUS client and RADIUS server before they can communicate. If the client and server are on the same network, one administrator may be able to configure both sides of the RADIUS communication. If the client and server are not administered by the same person, you may have to coordinate RADIUS configuration details with the administrators of other networks.

RADIUS Server Configuration

To configure Steel-Belted Radius to respond to RADIUS clients, run the SBR Administrator, open the RADIUS Clients panel, and enter the following information for each RADIUS client:
The IP address of the client device.

The RADIUS shared secret used by Steel-Belted Radius and the client device. For information on RADIUS shared secrets, see “Shared Secrets” on page 8.

The make and model of the client device, selected from a list of devices that Steel-Belted Radius supports. If a specific make and model is not listed, select - Standard Radius -.

Additionally, you must configure the UDP ports the server will use when sending and receiving RADIUS authentication and accounting packets. The UDP ports you configure on the RADIUS server must match the UDP ports that the RADIUS client is using for the same purposes. For more information, see “RADIUS Ports” on page 6.

**RADIUS Client Configuration**

You must tell each RADIUS client how to contact its RADIUS server. To configure a client to work with a Steel-Belted Radius server, log in to the client device, run its administration program, bring up its RADIUS configuration interface, and enter the following information:

- The IP address of the Steel-Belted Radius server.
- The RADIUS shared secret to be used by Steel-Belted Radius and the client device. For information on RADIUS shared secrets, see “Shared Secrets” on page 8.
- The UDP ports on which to send and receive RADIUS authentication and accounting packets. These must match the UDP ports that Steel-Belted Radius is using for the same purposes. For more information, see “RADIUS Ports” on page 6.

**Multiple RADIUS Servers**

You can distribute the RADIUS workload among several servers, as follows:

- You can set up separate servers for RADIUS authentication and accounting services. When RADIUS authentication and accounting services are performed by separate servers, each client device must be configured to send its authentication packets to one RADIUS server and its accounting packets to another.

- You can provide redundancy by pairing RADIUS servers to work in tandem. Most NAD configuration interfaces permit you to designate primary and secondary servers for authentication and accounting.

If both measures for distributing the RADIUS workload are implemented, client configuration involves identifying four servers for each client device: a primary RADIUS accounting server, a secondary RADIUS accounting server, a primary RADIUS authentication server, and a secondary RADIUS authentication server.
Shared Secrets

A shared secret is a case-sensitive text string used to validate communications between two RADIUS devices. You should configure shared secrets that are long enough and random enough to resist attack, and you should avoid using the same shared secret throughout your network. To maximize the security of your server’s shared secret, consider using Juniper Network’s free Password Amplifier utility, which takes an ordinary shared secret or password (swordfish) and hashes it repeatedly to produce a 16-character amplified secret (g8QvQuRgRsl1AQ1E). You can paste this amplified secret into your server configuration to maximize security.

NOTE: For more information on Juniper Network’s free Password Amplifier utility, see http://www.juniper.net/customers/support/products/aaa_802/sbr_demo.jsp.

Steel-Belted Radius uses three types of shared secrets:

- RADIUS secret – Used to authenticate communication between a RADIUS server and a RADIUS client
- Replication secret – Used to authenticate communication between a primary server and a replica server
- Node secret – If you use RSA SecurID, Steel-Belted Radius uses a node secret to authenticate communication between a RADIUS server and an RSA Authentication Manager server.

Figure 3: Shared Secrets
RADIUS Secret

A RADIUS shared secret is a case-sensitive password used to validate communications between a RADIUS server, such as Steel-Belted Radius, and a RADIUS client, such as a network access device. Steel-Belted Radius supports shared secrets of up to 127 alphanumeric characters, including spaces and the following special characters:

~!@#$%^&*()_+|\=-'{}[]:";'<>?/,.

Identical shared secrets must be configured on both sides of the RADIUS communication link.

**NOTE:** Not all network access devices support shared secrets of up to 127 alphanumeric/special characters. You should select shared secrets that are fully supported by RADIUS devices in your network.

Most RADIUS clients allow you to configure different secrets for authentication and accounting. On the server side, the configuration interface allows you to create a list of known RADIUS clients (network access devices). You should be able to identify the authentication shared secret and accounting shared secret that a server uses to communicate with each of the clients on this list.

During an authentication transaction, password information must be transmitted securely between the RADIUS client (network access device) and Steel-Belted Radius. Steel-Belted Radius uses the authentication shared secret to encrypt and decrypt password information.

No encryption is involved in transmitting accounting data between a RADIUS client and RADIUS server. However, the accounting shared secret is used by each device to verify that it can “trust” any RADIUS communications it receives from the other device.

Replication Secret

A replication secret is a text string used to authenticate communications between a primary server and a replica server. You do not need to configure the replication secret for a realm: the primary server generates it automatically, and each replica server in a realm receives the replication secret as part of its configuration package.

See “About Replication” on page 153 for information on primary and replica servers.

Node Secret

If you use Steel-Belted Radius with RSA SecurID, the RSA Authentication Manager software views the Steel-Belted Radius service as a host agent. You must configure a node secret to authenticate communication between Steel-Belted Radius and the RSA Authentication Manager. A node secret is a pseudorandom string known only to the Steel-Belted Radius and RSA Authentication Manager. Before the Steel-Belted Radius sends an authentication request to the RSA Authentication Manager, it encrypts the data using a symmetric node secret key.


**Accounting**

A NAD can issue an Accounting-Request whenever it chooses, for example upon establishing a successful connection. Each time an Accounting-Request message arrives at the Steel-Belted Radius server, an accounting transaction begins. During this transaction, the server handles the message by examining the Acct-Status-Type and other attributes within the message, and taking the appropriate action.

**Attributes**

You work with RADIUS attributes while setting up users, profiles, and RADIUS clients in Steel-Belted Radius. The SBR Administrator lets you select RADIUS attributes by name from a predefined list. For each attribute, the SBR Administrator prompts you to enter values using familiar data types such as string, integer, telephone number, or network address.

**Dictionaries**

Steel-Belted Radius uses *dictionary files* to store lists of RADIUS attributes. Steel-Belted Radius uses these dictionaries to parse authentication/accounting requests and generate responses.

The main Steel-Belted Radius dictionary file (radius.dct) lists attributes defined by the RADIUS standard. The radius.dct file resides in the same directory as the Steel-Belted Radius service/process (usually C:\Program Files\Juniper Networks\Steel-Belted Radius\Service on Windows computers or /opt/JNPRsbr/radius on Solaris/Linux computers).

**Vendor-Specific Attributes**

In addition to the standard attributes, many network access devices use Vendor-Specific Attributes (VSAs) to complete a connection. Steel-Belted Radius supports a large number of specific network access devices by providing vendor-specific, proprietary dictionary files. These files also reside in the server directory and use the filename extension .dct.

**Dictionaries and the Make/Model Field**

During Steel-Belted Radius configuration, when you make a selection in the RADIUS Client Make/Model field, you are telling the server which dictionary file contains the VSAs for this client device. Thereafter, whenever the server receives a RADIUS packet from this client device, it can consult this dictionary file for any nonstandard attributes that it encounters in the packet. Standard RADIUS attributes are always defined by the radius.dct file. If you are not sure which make/model you should specify for a RADIUS Client, choose - Standard RADIUS -.

The selections available in the Make/model field identify devices whose vendors have provided attribute dictionaries for use with Steel-Belted Radius.
Chapter 2: RADIUS Basics

Updating Attribute Information
If your NAD vendor announces a new product, a new attribute, or a new value for an attribute, you can add this information to your Steel-Belted Radius configuration. You can edit the dictionary file for that vendor to add new attributes or attribute values, or you can create a new vendor-specific dictionary file that contains new attributes and values. For more information on dictionary files, refer to the Steel-Belted Radius Reference Guide.

User Attribute Lists
Each user entry in the Steel-Belted Radius database provides the information necessary for the server to try to authenticate a connection request using a specific authentication method. When you view a user entry using the SBR Administrator program, this method is identified in the User type field.

You can control authentication at finer levels of detail than simple username/password checking allow. The checklist, return list, or profile fields in the user entry in the database provide powerful tools for the authentication and authorization of users. These fields tell the server how to handle RADIUS attributes while authenticating a connection request and can be used to configure the authorization of the session.

Checklist Attributes
A checklist is a set of attributes that must accompany the authentication request before the request can be accepted. The NAD must send attributes that match the checklist associated with a user entry; otherwise, Steel-Belted Radius rejects the user even if the user’s name and password are valid.

By including appropriate attributes in the checklist, a variety of rules can be enforced. For example, only specific users might be permitted to use ISDN or dial-in connections to a particular NAD, or Caller ID might be used to validate a user against a list of acceptable originating telephone numbers.

A checklist is created by selecting attributes from a list of all RADIUS attributes known to the Steel-Belted Radius server. This list can include a variety of vendor-specific attributes.

During authentication, Steel-Belted Radius filters the checklist based on the dictionary for the RADIUS client that sent the authentication request. The server ignores any checklist attribute that is not valid for this device.

Return List Attributes
A return list is a set of attributes that Steel-Belted Radius must return to the NAD after authentication succeeds. The return list usually provides additional parameters that the NAD needs to complete the connection, typically as part of PPP negotiations. Return list attributes can thus be considered to be “authorization configuration parameters.”

By including appropriate attributes in the return list, you can create a variety of connection policies. Specific users can be assigned particular IP addresses or IPX network numbers; IP header compression can be turned on or off; or a time limit can be assigned to the connection.
You create a return list by selecting attributes from a list of all RADIUS attributes known to Steel-Belted Radius. This list can include a variety of vendor-specific attributes.

During authentication, Steel-Belted Radius filters the return list based on the dictionary for the RADIUS client that sent the authentication request. The server omits any return list attribute that is not valid for this device.

**Attribute Values**

The value of each RADIUS attribute has a well-defined data type: numeric, string, IP or IPX address, time, or hexadecimal. For example, `Callback-Number` is of type `string` and contains a telephone number, while `NAS-Port-Type` is an item from a list, and can be `Sync`, `Async`, and so forth.

**NOTE:** Steel-Belted Radius supports signed integers (negative numbers) for attributes received in packets and processing relating to those attributes. However, SBR Administrator does not support signed integers, and treats signed and unsigned integers as unsigned integers.

**Single- and Multi-Valued Attributes**

Attributes can be single- or multi-valued. Single-valued attributes appear at most once in the checklist or return list; multi-valued attributes may appear several times.

If an attribute appears more than once in the checklist, this means that any one of the values is valid. For example, you can set up a checklist to include multiple telephone numbers for attribute `Calling-Station-ID`. A user trying to dial into your network would then have to call from one of the designated telephone numbers to be authenticated.

If an attribute appears more than once in the return list, each value of the attribute is sent as part of the response packet. For example, to enable both IP and IPX header compression for a user, the `Framed-Compression` attribute should appear twice in the return list: once with the value `VJ-TCP-IP-header-compression` and once with the value `IPX-header-compression`.

**Orderable Multi-Valued Attributes**

Certain multi-valued return list attributes are also orderable, which means the attribute can appear more than once in a RADIUS response, and the order in which the attributes appear is important.

For example, the `Reply-Message` attribute allows text messages to be sent back to the user for display. A multi-line message is sent by including this attribute multiple times in the return list, with each line of the message in its proper sequence.

**System Assigned Values**

Some attributes do not allow the administrator to set a value. Steel-Belted Radius retrieves the appropriate value for this attribute when it is needed.
Echo Property

Using the echo property, you can force an attribute from the RADIUS request to be echoed in the RADIUS response. For example, you might add Callback-Number to the return list and select the echo check box. Steel-Belted Radius takes the value of the Callback-Number it receives in the RADIUS request and echoes it back to the client in the RADIUS response; if it receives no Callback-Number, it echoes nothing.

You enter Callback-Number one or more times into the checklist. This indicates that one of the callback numbers you supplied must be present in the RADIUS request, and that number should be echoed in the RADIUS response.

Default Values

Selecting default for a checklist attribute specifies that, if the RADIUS request does not include this attribute, the request should not be rejected. Instead, the value supplied as the default should be used as if it were received as part of the request. One use for default values is to require that an attribute in a RADIUS request must have one of several values, or must not be present at all. Another use is to provide a default value for an attribute in conjunction with the echo property in the return list.

Steel-Belted Radius can provide alternate values when an attribute appears in the checklist marked as default, and the same attribute appears in the return list marked as echo. The server echoes the actual value of the attribute in the RADIUS response if the attribute appears in the RADIUS request and echoes the default value (from the checklist) in the response if the attribute does not appear in the RADIUS request.

If you add multiple values of the same attribute to the checklist, only one of them can be marked as default.

For example, an administrator adds several Callback-Number values to the checklist and marks one of them as default. The administrator adds Callback-Number to the return list and specifies it as echo.

- If a Callback-Number value is present in the RADIUS request, it must match one of the checklist values or the user is rejected.
- If it does match, the user is accepted and the value supplied is echoed in the RADIUS response.
- If no Callback-Number is supplied in the request, the user is accepted and the default value is echoed in the response.

Other checklist attributes are used to provide configuration for the user, such as time-of-day and concurrent-login-limit information.
Centralized Configuration Management

Steel-Belted Radius supports the replication of RADIUS configuration data from a primary server to a maximum of 10 replica servers within a replication realm. Replica servers help balance the load of authentication requests coming in from RADIUS clients, and ensure that authentication services are not interrupted if the primary or other replica servers stops working.

For example, Figure 4 illustrates an environment where RADIUS traffic is load-balanced by configuring each network access device to authenticate users through a different RADIUS server (solid line). If a RADIUS server becomes unavailable, the NAD can fail over to its backup RADIUS server (dotted line).

**Figure 4: Using Replication for Redundancy and Load Balancing**

All the servers within a realm reflect the current configuration specified by the network administrator: the network administrator modifies the configuration on the primary server, and the primary server propagates the new configuration to its replica servers. For example, after a network administrator configures a new RADIUS client or profile on the primary server, the network administrator tells the primary server to publish a configuration package file (replica.ccmpkg) that contains the updated configuration information. After publication, the primary server notifies each replica server that a new configuration package is ready. Each replica then downloads and installs the configuration package to update its settings.
Chapter 2: RADIUS Basics

Figure 5: Configuration Package Publication

The primary server maintains a list of the replica servers that have registered with it. The primary server uses this list to track which servers to notify after it publishes an updated configuration package to resynchronize the configuration of replica servers.

If the primary server needs to be taken out of service, the network administrator promotes one of the replica servers to be the new primary server. Thereafter, the other replica servers copy the configuration package from the promoted primary server.

**Proxy RADIUS**

The Steel-Belted Radius server can forward a RADIUS request to another server for processing and relay the other server’s result back to its client. In such cases, Steel-Belted Radius is acting as a *proxy* for the target server, and Steel-Belted Radius is *proxy-forwarding* the request to the target server.

Steel-Belted Radius supports proxy RADIUS; any Steel-Belted Radius server can act as proxy or target for authentication or accounting messages.

**Proxy RADIUS Authentication**

RADIUS authentication messages are proxy-forwarded as follows:

1. An access client requests authentication from a RADIUS client, which sends an authentication request to a RADIUS proxy server.

2. The proxy RADIUS server forwards the message to a RADIUS *target* server.

3. The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.

4. The proxy RADIUS server relays the acknowledgement response message to the RADIUS client.
Figure 6: RADIUS Proxy Forwarding

Proxy RADIUS Accounting

RADIUS accounting messages are proxy-forwarded as follows:

1. A RADIUS server receives an accounting request.

2. Depending on its configuration, the RADIUS proxy server forwards the accounting message to a target accounting server or records accounting attributes locally (or does both).

3. If the proxy server does not receive an acknowledgement of the forwarded accounting message, it re-sends periodically according to its retry policy.

4. When the target server acknowledges the request, the proxy server forwards an acknowledgement to the RADIUS client.

Authentication

RADIUS uses different types of messages during user authentication. Table 1 summarizes the conditions under which each type of RADIUS authentication message is issued, and the purpose of any RADIUS attributes the message contains.

Table 1: RADIUS Authentication Messages and Attributes

<table>
<thead>
<tr>
<th>Message Conditions</th>
<th>Purpose of Message Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>When a NAD receives a connection request from a user, the NAD requests authentication by sending an Access-Request to its RADIUS server.</td>
<td>Identify the user. Describe the type of connection the user is trying to establish.</td>
</tr>
<tr>
<td>When a RADIUS server is able to authenticate a user, it returns a RADIUS Access-Accept to the NAD.</td>
<td>Allow the NAD to complete access negotiations. Configure connection details such as providing the NAD with an IP address it can assign to the user. Enforce time limits and other “class of service” restrictions on the connection.</td>
</tr>
<tr>
<td>When a RADIUS server is unable to authenticate a connection request, it returns an Access-Reject to the NAD.</td>
<td>Terminate access negotiations. Identify the reason for the authorization failure.</td>
</tr>
</tbody>
</table>
Chapter 2: RADIUS Basics

Authentication Methods

Each time an Access-Request message reaches the server, an authentication transaction begins. During this transaction, the server attempts to authenticate the request by sequentially trying its configured and enabled authentication methods. The server consults its list of authentication methods to determine which methods to try and in which order to try them.

Native User Authentication

Native user authentication references user accounts stored on the Steel-Belted Radius server. When trying the native user method, Steel-Belted Radius searches its database for an entry whose User-Type is Native User, and whose username matches the User-Name in the Access-Request.

- If the entry cannot be found, or if it is found and the password is invalid, Steel-Belted Radius tries the next enabled method in the authentication methods list.
- If an entry for the user is found but the entry’s checklist does not match attributes found in the Access-Request, Steel-Belted Radius returns an Access-Reject message to the NAD.
- If the entry is found and its password and checklist match perfectly, Steel-Belted Radius formats an Access-Accept message using the entry’s return list, and returns it to the NAD.

Pass-Through Authentication

Pass-through authentication methods permit Steel-Belted Radius to begin the authentication by asking another entity to validate the username and password found in the Access-Request.

Steel-Belted Radius can pass authentication requests through to a Windows security database, RSA Authentication Manager (RSA SecurID), or TACACS+ server.

Proxy RADIUS Authentication

Steel-Belted Radius can convey an Access-Request to some other RADIUS server, which then (1) attempts to authenticate the connection request according to its own conventions and (2) returns a response to Steel-Belted Radius. Steel-Belted Radius then relays this response to the NAD. The set of conventions for relaying packets between cooperating RADIUS servers is known as proxy RADIUS.
External Authentication

External authentication methods enable Steel-Belted Radius to authenticate users by referring to external SQL or LDAP databases. During external authentication, Steel-Belted Radius queries the database for authentication data, and uses the results to format a response packet. Steel-Belted Radius then relays this response to the NAD.

For information on using Steel-Belted Radius with SQL databases, see Chapter 15, “Configuring SQL Authentication” on page 193. For information on using Steel-Belted Radius with LDAP databases, see Chapter 17, “Configuring LDAP Authentication” on page 219.

Directed Authentication (GEE/SPE only)

Every authentication request works its way through the same Authentication Methods list until one of the methods succeeds or the end of the list is reached.

This behavior might not be ideal for every user (GEE) or account (SPE). If you want requests from certain users or accounts to bypass the master Authentication Methods list and use an alternate list, you can do so by employing the directed authentication feature. This feature allows you to map the User-Name or DNIS information in an incoming authentication request to a specific list of authentication methods. The list can include any native, pass-through, proxy-as-authentication, or external database authentication method configured on the Steel-Belted Radius server.

You can also direct authentication towards a particular realm using a technique called attribute mapping. This allows you to check for the presence or absence of a particular attribute in an authentication request, or for an attribute containing a specific value. Attribute mapping can be used with both proxy realms and directed realms.

HTTP Digest Access Authentication

HTTP Digest Access authentication provides a simple challenge-response authentication mechanism that an HTTP server can use to challenge an HTTP client request.

Steel-Belted Radius supports two forms of HTTP Digest Access authentication:

- HTTP Digest Access authentication, which is described in draft-ietf-radext-digest-auth-05.txt, uses 13 new RADIUS attributes to authenticate access requests from an HTTP server. When HTTP Digest Access authentication is used:
  a. An HTTP client sends a request without an authorization header to an HTTP server.
  b. The HTTP server sends a challenge containing a random value (nonce) to the HTTP client.
  c. The HTTP client creates an MD5 hash containing the username, password, nonce value, and other information, and returns this MD5 hash to the HTTP server in a request with an authentication header.
d. The HTTP server sends an Access-Request message containing special RADIUS attributes to Steel-Belted Radius.

e. Steel-Belted Radius verifies the HTTP client’s credentials and returns a RADIUS Access-Accept or Access-Reject message to the HTTP server.

The Ericsson ViG version of HTTP Digest Access authentication, which is described in draft-sterman-aaa-sip-05.txt, uses two Ericsson vendor-specific attributes (a Digest-Response attribute and one or more Digest-Attributes attributes) to authenticate access requests. When Ericsson ViG HTTP Digest Access authentication is enabled, Steel-Belted Radius looks for the ViG VSAs when it parses incoming packets, and, if it finds them, converts them to AVPs compatible with HTTP Digest Access authentication.

You must edit settings in the radius.ini file to enable HTTP Digest Access authentication.

**Authenticate-Only Requests**

Steel-Belted Radius supports requests to authenticate a user where the server performs no other processing. The NAD specifies this type of request by setting the Service-Type field to a value of Authenticate-Only (numeric value 8). The server responds with either an Access-Reject or an Access-Accept (without any attributes).

You can disable this feature (so that attributes are always returned in the response packet) by setting the AuthenticateOnly field in the [Configuration] section of the radius.ini file to 0. For more information on radius.ini, refer to the Steel-Belted Radius Reference Guide.

**Configuring the Authentication Sequence**

After you configure authentication methods for Steel-Belted Radius, the Authentication Policies panel in the SBR Administrator displays them in the order in which the server tries them. Enabled methods are displayed in black text; disabled methods are displayed in gray text. During an authentication transaction, the server works down the list, skipping disabled methods.

You can enable or disable methods or re-order methods in the list by using the controls in the Authentication Methods tab of the Authentication Policies panel. For information on setting up authentication sequences, see Chapter 12, “Setting Up EAP Authentication Policies” on page 105.

**Configuring Authentication Methods**

Each authentication method in Steel-Belted Radius performs a different type of processing on information in an incoming Access-Request packet. Table 2 summarizes what you need to do to configure each authentication method.

<table>
<thead>
<tr>
<th>Method</th>
<th>How to Configure</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native User</td>
<td>Create native user entries in the Steel-Belted Radius database.</td>
<td>“Setting Up Native Users” on page 48</td>
</tr>
</tbody>
</table>
Table 2: Authentication Method Configuration (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>How to Configure</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Pass-Through Security</td>
<td>This method assumes that you already have users, groups, and passwords defined in your local security database. Create user entries in the Steel-Belted Radius database. Choose User-types as appropriate.</td>
<td>Chapter 5, “Administering Users” on page 47</td>
</tr>
<tr>
<td>RSA SecurID</td>
<td>This method assumes that you already have PIN/token code pairs defined on an RSA SecurID server. First, configure Steel-Belted Radius to communicate with the RSA SecurID server. Then create user entries in the Steel-Belted Radius database. Choose SecurID User, &lt;ANY&gt;, SecurID Prefix, and SecurID Suffix user-types.</td>
<td>“Setting Up SecurID Users” on page 58</td>
</tr>
<tr>
<td>TACACS+</td>
<td>This method assumes that you have username/password pairs defined on a TACACS+ server. First, configure Steel-Belted Radius to communicate with the TACACS+ server. Then, create user entries in the Steel-Belted Radius database. Assign TACACS+ User, &lt;ANY&gt;, TACACS+ Prefix, and TACACS+ Suffix user-types.</td>
<td>“Setting Up TACACS+ Users” on page 60</td>
</tr>
<tr>
<td>Proxy RADIUS</td>
<td>Add a single target. You can set up single targets that are not associated with any realm.</td>
<td>Chapter 7, “Administering Proxy RADIUS” on page 71</td>
</tr>
<tr>
<td>EAP-TTLS</td>
<td>This method provides a means for an authentication request to be sent directly from the client to the server through a TLS connection. The act of establishing the TLS connection authenticates the server to the client and the authentication request sent through the tunnel authenticates the client to the server. Create a Steel-Belted Radius ttlsauth.aut file that specifies options for the TLS connection and the manner in which Steel-Belted Radius routes the inner authentication request. Stop and restart Steel-Belted Radius. Subsequently, the EAP-TTLS authentication method appears in the Authentication Methods tab in the Authentication Policies panel. You can use the Authentication Policies panel to enable, disable, and re-order EAP-TTLS methods.</td>
<td>“EAP-TTLS” on page 124</td>
</tr>
<tr>
<td>External SQL Database</td>
<td>This method assumes that you have user records stored in a SQL database. Create a Steel-Belted Radius .aut file that connects to a SQL database and issues a SELECT query based upon the username and password. Give the .aut file a unique InitializationString value. Stop and restart Steel-Belted Radius. Subsequently, the SQL authentication method appears in the Authentication Methods tab of the Authentication Policies panel, using the InitializationString value as its name. You can use the Authentication Policies panel to enable, disable, and re-order the SQL authentication method.</td>
<td>“Configuring SQL Authentication” on page 196</td>
</tr>
</tbody>
</table>
Two-Factor Authentication

Extensible Authentication Protocol-Tunneled Transport Layer Security (EAP-TTLS) provides for certificate-based mutual authentication between a client and a network through an encrypted tunnel. A typical implementation of EAP-TTLS uses certificates on authentication servers to create a network-to-user encryption tunnel, and then uses EAP inside the TLS tunnel for user-to-network authentication.

An enhanced version of EAP-TTLS uses certificates on the client side to provide two-factor authentication: the end user must have both a private key for a valid certificate and the password to an active account to obtain network access.

When client certificate support in EAP-TTLS is enabled on the server, you must provide a list of trusted root certificates from which offered client certificates must derive. These certificates must be provided in DER-encoded form and must be placed in the root subdirectory of the server directory.

Optionally, you can enable certificate revocation list (CRL) checking as part of the EAP-TTLS authentication process. CRL checking verifies that an unexpired certificate has not been revoked by its issuing Certificate Authority (CA) for any reason, such as a suspected security breach. Enabling CRL checking means that, every time the client requests a connection, Steel-Belted Radius checks the CRL to confirm that the client certificate has not been revoked. This improves security but increases processing overhead.

Note that, if client certificate support is not enabled in EAP-TTLS, any trusted root certificates and CRL checking options are ignored.

Table 2: Authentication Method Configuration (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>How to Configure</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>External LDAP Database</td>
<td>This method assumes that you have user records stored in an LDAP database. Create a Steel-Belted Radius .aut file that validates the username and password based upon Bind and Search requests to an LDAP database. Give the .aut file a unique InitializationString value. Stop and restart Steel-Belted Radius. Subsequently, the LDAP authentication method appears in the Authentication Methods tab of the Authentication Policies panel, using the InitializationString value as its name. You can use the Authentication Policies panel to enable, disable, and re-order the LDAP authentication method as desired.</td>
<td>“Configuring LDAP Authentication” on page 222</td>
</tr>
</tbody>
</table>

**Two-Factor Authentication**

Extensible Authentication Protocol-Tunneled Transport Layer Security (EAP-TTLS) provides for certificate-based mutual authentication between a client and a network through an encrypted tunnel. A typical implementation of EAP-TTLS uses certificates on authentication servers to create a network-to-user encryption tunnel, and then uses EAP inside the TLS tunnel for user-to-network authentication.

An enhanced version of EAP-TTLS uses certificates on the client side to provide two-factor authentication: the end user must have both a private key for a valid certificate and the password to an active account to obtain network access.

When client certificate support in EAP-TTLS is enabled on the server, you must provide a list of trusted root certificates from which offered client certificates must derive. These certificates must be provided in DER-encoded form and must be placed in the root subdirectory of the server directory.

Optionally, you can enable certificate revocation list (CRL) checking as part of the EAP-TTLS authentication process. CRL checking verifies that an unexpired certificate has not been revoked by its issuing Certificate Authority (CA) for any reason, such as a suspected security breach. Enabling CRL checking means that, every time the client requests a connection, Steel-Belted Radius checks the CRL to confirm that the client certificate has not been revoked. This improves security but increases processing overhead.

Note that, if client certificate support is not enabled in EAP-TTLS, any trusted root certificates and CRL checking options are ignored.

Table 2: Authentication Method Configuration (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>How to Configure</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>External LDAP Database</td>
<td>This method assumes that you have user records stored in an LDAP database. Create a Steel-Belted Radius .aut file that validates the username and password based upon Bind and Search requests to an LDAP database. Give the .aut file a unique InitializationString value. Stop and restart Steel-Belted Radius. Subsequently, the LDAP authentication method appears in the Authentication Methods tab of the Authentication Policies panel, using the InitializationString value as its name. You can use the Authentication Policies panel to enable, disable, and re-order the LDAP authentication method as desired.</td>
<td>“Configuring LDAP Authentication” on page 222</td>
</tr>
</tbody>
</table>

**Two-Factor Authentication**

Extensible Authentication Protocol-Tunneled Transport Layer Security (EAP-TTLS) provides for certificate-based mutual authentication between a client and a network through an encrypted tunnel. A typical implementation of EAP-TTLS uses certificates on authentication servers to create a network-to-user encryption tunnel, and then uses EAP inside the TLS tunnel for user-to-network authentication.

An enhanced version of EAP-TTLS uses certificates on the client side to provide two-factor authentication: the end user must have both a private key for a valid certificate and the password to an active account to obtain network access.

When client certificate support in EAP-TTLS is enabled on the server, you must provide a list of trusted root certificates from which offered client certificates must derive. These certificates must be provided in DER-encoded form and must be placed in the root subdirectory of the server directory.

Optionally, you can enable certificate revocation list (CRL) checking as part of the EAP-TTLS authentication process. CRL checking verifies that an unexpired certificate has not been revoked by its issuing Certificate Authority (CA) for any reason, such as a suspected security breach. Enabling CRL checking means that, every time the client requests a connection, Steel-Belted Radius checks the CRL to confirm that the client certificate has not been revoked. This improves security but increases processing overhead.

Note that, if client certificate support is not enabled in EAP-TTLS, any trusted root certificates and CRL checking options are ignored.
Password Protocols

During an authentication transaction, password information is transmitted between the NAD and the RADIUS server. This password information originally comes from the user, for example during PPP negotiations between a user and a NAD. Steel-Belted Radius supports four protocols (PAP, CHAP, MS-CHAP, and MS-CHAP-V2) for receiving the password from the NAD. Steel-Belted Radius also supports the Extensible Authentication Protocol.

Table 3 lists supported protocols according to the authentication methods with which each protocol can be used.

Table 3: Authentication Methods and Password Protocols

<table>
<thead>
<tr>
<th>Method</th>
<th>PAP</th>
<th>CHAP</th>
<th>MS-CHAP</th>
<th>MS-CHAP-V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>Yes</td>
<td>Yes, if BindName is used and the password is in clear text form or is encrypted with enc-md5.</td>
<td>Yes, if BindName is used and the password is in clear text form or is encrypted with enc-md5.</td>
<td>Yes, if the LDAP server can return clear-text password or MD4 hash of Unicode form of password.</td>
</tr>
<tr>
<td>No, if Bind is used</td>
<td>No, if Bind is used</td>
<td>No, if Bind is used</td>
<td>No, if Bind is used</td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Domain Group</td>
<td>Yes</td>
<td>No</td>
<td>Yes, if the user is in local or trusted domain</td>
<td>Yes, if the user is in local or trusted domain</td>
</tr>
<tr>
<td>Windows Domain User</td>
<td>Yes</td>
<td>No</td>
<td>Yes, if the user is in local or trusted domain</td>
<td>Yes, if the user is in local or trusted domain</td>
</tr>
<tr>
<td>Proxy RADIUS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SecurID</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL</td>
<td>Yes</td>
<td>Yes, if the password is available in clear text form in the database or is encrypted with enc-md5.</td>
<td>Yes, if the password is available in clear text form in the database or is encrypted with enc-md5.</td>
<td>Yes, can return clear-text password or MD4 hash of Unicode form of password.</td>
</tr>
<tr>
<td>TACACS+</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>UNIX User</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>UNIX Group</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Password Authentication Protocol

When the Password Authentication Protocol (PAP) is used, a remote user negotiates with the NAD “in the clear,” and no encryption is used to send the password to the NAD. After the NAD has enough information from the user to create an Access-Request, the NAD encrypts the password (using its RADIUS shared secret) before sending an Access-Request packet to Steel-Belted Radius.
Upon receiving the Access-Request, Steel-Belted Radius looks for attributes within the packet that identify the NAD that sent it. Steel-Belted Radius decrypts the password by using the shared secret configured for the RADIUS client entry associated with the sending NAD.

Ultimately, Steel-Belted Radius has the password in clear text form for authentication.

**Challenge Handshake Authentication Protocol**

The Challenge Handshake Authentication Protocol (CHAP) avoids sending passwords in clear text over any communication link. Under CHAP, during password negotiations the NAD generates a *challenge* (a random string) and sends it to the user. The user’s PPP client creates a *digest* (the password concatenated with the challenge), encrypts the digest using one-way encryption, and sends the digest to the NAD.

The NAD sends this digest as the password in the Access-Request.

Because the encryption is one-way, Steel-Belted Radius cannot recover the password from the digest. Instead, it performs an identical operation, using the NAD’s challenge value (provided in the Access-Request packet) and its own copy of the user’s password to generate its own digest. If the two digests match, the password is the same.

Steel-Belted Radius must be able to perform the digest operation to support CHAP. Therefore, it must have access to its own copy of the user’s password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password by means of a query to the database; the retrieved password can be used to create a digest if it is in clear text form. A TACACS+ server provides CHAP support and handles the digest operation itself after Steel-Belted Radius sends the username and password through. No other authentication methods support CHAP at this time.

**MS-CHAP and MS-CHAP-V2**

The two varieties of MS-CHAP (Microsoft Challenge Handshake Authentication Protocol) are Microsoft authentication protocols that, like CHAP, avoid sending passwords in clear text. Steel-Belted Radius supports both 40-bit and 128-bit MS-CHAP methods. Steel-Belted Radius must be able to perform a digest operation similar to CHAP to support MS-CHAP. Therefore, it must have access to its own copy of the user’s password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password by means of a query to the database; the retrieved password can be used to create a digest if it is in clear text form.

MS-CHAP and MS-CHAP-V2 communicate users’ requests to change their passwords to a RADIUS server. Steel-Belted Radius supports this feature, although it must also be supported by whatever application the user is using to log in.

MS-CHAP and MS-CHAP-V2 operate in the same way, but they use different attributes. An MS-CHAP client won’t accept MS-CHAP-V2 attributes, and vice-versa; be careful to use the appropriate set of attributes.
Accounting


To understand the Steel-Belted Radius accounting sequence, you will need an overview of RADIUS accounting messages. Table 4 explains the conditions under which each type of message is issued, and the purpose of any RADIUS attributes that a message contains.

Table 4: Message Conditions and Attributes

<table>
<thead>
<tr>
<th>Message Conditions</th>
<th>Purpose of Message Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The RADIUS client sends accounting data to Steel-Belted Radius using an Accounting-Request message. The RADIUS client is responsible for verifying that the server receives accounting requests. Most clients retry periodically until the server responds.</td>
<td>Depending on the value of the Acct-Status-Type attribute, the message type is considered to be Start, Stop, Interim-Acct, Accounting-On, or Accounting-Off.</td>
</tr>
<tr>
<td>Upon receipt of an Accounting-Request message, the server sends an Accounting-Response.</td>
<td>Complete the request/response cycle.</td>
</tr>
<tr>
<td>After receiving an Access-Accept from the server, the NAD completes its access negotiation with the user. The NAD then sends a Start message to the server.</td>
<td>Record connection data, such as username, NAD identifier, NAD port identifier, port type, and connection start time.</td>
</tr>
<tr>
<td>At intervals of approximately every six minutes, the NAD sends an Interim-Acct message to the server.</td>
<td>Record a “snapshot” of statistics regarding the connection. One message contains the current value of every statistic that this NAD is capable of recording about this type of connection.</td>
</tr>
<tr>
<td>After a connection is terminated, the NAD sends a Stop message to the server.</td>
<td>Record statistics regarding the connection. One message contains the final value of every statistic that this NAD is capable of recording about this type of connection.</td>
</tr>
<tr>
<td>Every time a client device comes online, whether after a crash or after an orderly shutdown, it sends an Accounting-On message to the server.</td>
<td>Identify the device that is going online and clear all session information.</td>
</tr>
<tr>
<td>Every time a client device experiences an orderly shutdown, before completing its shutdown sequence it sends an Accounting-Off message to the server.</td>
<td>Identify the device that is going offline and clear all session information.</td>
</tr>
</tbody>
</table>

**Accounting Sequence**

A NAD can issue an Accounting-Request whenever it chooses, for example upon establishing a successful connection. Each time an Accounting-Request message reaches Steel-Belted Radius, an accounting transaction begins. During this transaction, the server handles the message by examining the Acct-Status-Type and other attributes within the message, and taking the appropriate action.
**Comma-Delimited Log Files**

When the Steel-Belted Radius accounting log is enabled, all of the RADIUS accounting attributes that the server receives are reformatted and logged to a comma-separated value (CSV) text file, which is easily imported into spreadsheets and database programs for report generation and billing.

**External Accounting**

External accounting methods permit Steel-Belted Radius to record accounting data to external databases. Configuration files specify how Steel-Belted Radius communicates with an external database and how to insert accounting data into that database.

SQL is the only external accounting method currently supported by Steel-Belted Radius.

See “About SQL Accounting” on page 207.

**Tunneled Accounting**

During authentication, a user is typically identified by attributes such as User-Name (in the authentication request) and Class (in the authentication accept response). Standard RADIUS accounting requests typically include these attributes in messages flagging Start, Interim, and Stop events so that the user’s identity can be recorded for accounting and auditing purposes.

When an organization uses a tunneled authentication protocol such as EAP/TTLS or EAP/PEAP, the identity of a user requesting authentication may be concealed from the NAD; the User-Name attribute carried by the outer authentication protocol is typically a non-unique value such as “anonymous.” As a result, the outer User-Name value included in accounting requests may not be sufficient to determine a user’s identity. Class attributes provided by an authentication server cannot be included in cleartext in an outer Access-Accept message because they might contain clues about the user’s identity, thereby defeating the identity-hiding feature of the tunneled protocol.

Tunneled accounting allows Steel-Belted Radius to pass user identity information to accounting processes without exposing user identities to a NAD that should not see them. When tunneled accounting is enabled, RADIUS attributes are encrypted and encapsulated in a Class attribute. If the information for a Class attribute exceeds the attribute payload size (253 octets), Steel-Belted Radius returns more than one Class attribute for a user.

The tunneled accounting transaction sequence is:

1. The Steel-Belted Radius server acting as the tunnel endpoint for EAP/TTLS or EAP/PEAP encrypts a user’s inner User-Name and Class attributes when it authenticates the user.

2. The server returns the encrypted information to the NAD encapsulated in a Class attribute in the outer Access-Accept message. The NAD associates this encapsulated identity attribute with the user, and echoes the encapsulated identity attribute whenever it generates an accounting request for the user.
3. When Steel-Belted Radius receives an accounting request from a network access device, it scans the request for an encapsulated identity attribute.

4. If Steel-Belted Radius finds an encapsulated identity attribute, it de-encapsulates and decrypts the attributes to reconstitute the original inner User-Name and Class attributes.

5. Steel-Belted Radius substitutes the decrypted attributes for the ones returned from the NAD.

6. Steel-Belted Radius processes the accounting request locally or forwards the accounting request through the proxy to its intended target.

To implement tunneled accounting, you must configure the classmap.ini file to specify how attributes should be presented, and you must configure the spi.ini file to specify the keys that are used to encrypt and decrypt users’ identity information. The classmap.ini file and the spi.ini file are described in the Steel-Belted Radius Reference Guide.

For an overview of how EAP/TTLS and EAP/PEAP work, refer to “About the Extensible Authentication Protocol” on page 105.

---

RADIUS Client Groups

If your RADIUS clients use the same RADIUS attributes and have contiguous IP addresses, you can configure one or more RADIUS client groups and specify an address range consisting of as many as 500 IP addresses for each client group. When Steel-Belted Radius receives a RADIUS request that includes a source IP address in this range, it uses the RADIUS client group to determine the appropriate shared secret, make/model, and IP address pool.

Please note the following when you set up address ranges for RADIUS client groups:

- Address ranges are for IPv4 networks only. Steel-Belted Radius does not support address ranges for IPv6 or IPX.

- The address range assigned to one RADIUS client group cannot overlap the address ranges assigned to other RADIUS client groups.

- The starting address of the address range assigned to a RADIUS client group cannot match the IP address of an individual RADIUS client.

- If an individual RADIUS client entry has an IP address that falls within an address range assigned to a RADIUS client group, Steel-Belted Radius uses the make/model for the individual RADIUS client. For example, if RADIUS client RAS1 is configured with IP address 192.168.21.55 and if RADIUS client group BLDG1RAS is configured with an IP address range 192.168.21.50–192.168.21.60, Steel-Belted Radius uses the client information for RAS1 if it receives a RADIUS request from 192.168.21.55, and it uses the client information for BLDG1RAS if it receives a RADIUS request from 192.168.21.56.
- A RADIUS client group cannot use a Class D, E, or F IP address (that is, an address greater than 223.255.255.0).

See “Adding a RADIUS Client or Client Group” on page 42 for information on how to configure IPv4 address ranges for RADIUS clients.

---

**IP Address Assignment**

Steel-Belted Radius can assign IP addresses to users in several ways:

- **Static assignment**—The same IP address is assigned to a user each time the user connects. For example, if the user Kevin has a `Framed-IP-Address` attribute set to 123.11.245.123, then the IP address 123.11.245.123 is assigned each time Kevin connects to the network.

- **Assignment from a specific address pool**—An address is assigned from a specific pool when the user connects. For example, if user Kevin has a `Framed-IP-Address` attribute set to the `Sales` IP address pool, the next available IP address from `Sales` is assigned when Kevin connects to the network.

- **Assignment from the RADIUS client’s IP address pool (or set of IP address pools)**—An address is assigned from one of the pools associated with the RADIUS client that makes the connection when a user connects. For example, assume that a RADIUS client called RAS1 uses IP address pool A, and a RADIUS client called RAS2 uses IP address pool B. A User entry called Kevin has a `Framed-IP-Address` attribute value of `pool associated with RADIUS Client`. When user Kevin gets a port on RAS1, an IP address from pool A is assigned. On the next call, Kevin might connect to RAS2; in this case an address from pool B is assigned.

Alternatively, if a user has been associated with a particular NAD-specific IP address pool (and suffix), an IP address from that pool is assigned.

---

**Address Pools and Replication**

Address pool information is not distributed with other configuration information in a replicated environment. If you are using IPv4 or IPX address pools in a replication environment, you must configure address pools separately on each replica server, and then use the same names to configure a master list of address pools on your primary server.

The master list of address pools configured on the primary server must include the names of all the pools on all of the replica servers. For example, Figure 7 illustrates a simple environment that uses four address pools. POOL1 and POOL2 are configured on one replica server and POOL4 is configured on a different replica server. As a consequence, the IP address pool list on the primary server must include POOL1, POOL2, POOL3 (the pool used by the primary server), and POOL4.
The network administrator must configure RADIUS clients (including the address pool associated with a RADIUS client) on the primary server. This RADIUS client/address pool association (but not the address pool information itself) is stored as part of the replication package passed from the primary server to the replica servers.

**Hints**

Steel-Belted Radius can treat the attribute **Framed-IP-Address** as a *hint*. This means that if this attribute appears in the Access-Request and the user return list is configured to allocate **Framed-IP-Address** from a pool, the IP address in the Access-Request is returned instead of the newly-allocated IP address.

This functionality is defined in the [Configuration] section of `radius.ini`:

```ini
[Configuration]
FramedIPAddressHint = <yes/no>
```

When hints are enabled, Steel-Belted Radius uses a hint to determine the value of the Framed-IP-Address attribute in the access response. This means that **Framed-IP-Address** in the Access-Request is returned in the Access-Accept, regardless of the **Framed-IP-Address** value stored in the user’s account.

The default value is `no`. 
Table 5 details the effect of hints:

<table>
<thead>
<tr>
<th>Account Configuration</th>
<th>Framed-IP-Address returned without hints</th>
<th>Framed-IP-Address returned with hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Framed-IP-Address</td>
<td>No value</td>
<td>Framed-IP-Address from Access-Request</td>
</tr>
<tr>
<td>Static Address</td>
<td>Static address</td>
<td>Static address</td>
</tr>
<tr>
<td>Address from Pool</td>
<td>Next address from pool</td>
<td>Framed-IP-Address from Access-Request</td>
</tr>
</tbody>
</table>

**NOTE:** By using hints, you can assign the same IP address to multiple active accounts.

---

**Resource Management**

This section explains how Steel-Belted Radius manages limited resources, such as network addresses, user or tunnel connections, and UDP ports.

**Network Address Assignment**

The Steel-Belted Radius address pooling feature allows you to set up one or more pools out of which unique network addresses are assigned dynamically as users require them. Each pool consists of a list of one or more ranges of IP addresses (an IP pool) or IPX network numbers (an IPX pool).

By using this feature, you can avoid allocating specific fixed addresses to individual users. You can make fewer addresses go farther, and you can consolidate address assignment across all your network access devices.

**How Address Assignment Works**

Proper operation of address assignment from a pool depends crucially on both RADIUS authentication and RADIUS accounting transactions, as follows:

1. During the RADIUS authentication transaction, if the user’s attribute settings specify address assignment from a pool, an address is allocated for that user from that pool.

2. The address is reserved for that user until a RADIUS accounting transaction indicates that the user has terminated the connection.

For this reason, the network access device must be configured for RADIUS accounting, and the same Steel-Belted Radius server must be specified for both authentication and accounting. If your NAD is not configured for accounting (or does not support accounting), you cannot use the address pooling feature because addresses would be assigned but never released.
Setting Return List Attributes
The Framed-IP-Address (or Framed-IPX-Address) return list attribute controls how the user’s IP (or IPX) address is assigned. The Framed-IP-Address or Framed-IPX-Address attribute can be set for each user in the Steel-Belted Radius database.

Handling Address Leaks
Under optimal conditions, Steel-Belted Radius assigns and releases addresses automatically. In some circumstances, you can get address leakage, where an address remains reserved for a user after the user has terminated the connection.

Address leakage occurs when the address has been assigned during the authentication transaction, but the accounting transaction that would have released the address is never received by Steel-Belted Radius. This can occur for several reasons:

- The Steel-Belted Radius server might have been taken down for a period of time during which accounting transactions occurred.
- The network access device might have failed or been taken down before the user terminated the connection. (In many cases, however, Steel-Belted Radius might be able to prevent address leakage by recovering the addresses when the NAD starts up again.)
- The network access device might have sent the authentication and accounting transactions to different RADIUS servers.
- Despite a successful authentication, the user’s PPP negotiation with the NAD might have terminated unsuccessfully for a variety of reasons. In such a case, some network access devices might not initiate a subsequent accounting transaction.
- Routing problems might have prevented the accounting transaction from reaching Steel-Belted Radius.

An address that has “leaked” remains out of circulation until you manually release it by displaying the Sessions list and deleting the corresponding session. See “Deleting Entries from the Sessions List” on page 242.

Address Leakage Upon Stopping and Starting the Server
Steel-Belted Radius maintains all current address assignments in a persistent database on disk. If you shut down the server and then restart it, all the information about which address is assigned to which user is retained.

Note that if you leave Steel-Belted Radius turned off for a substantial period of time after addresses have been assigned, address leakage may occur. After you restart the server, review the Current Sessions list (described in “Displaying the Current Sessions List” on page 239) and delete entries you know are obsolete.
Overlapping Address Ranges
If you maintain multiple IP or IPX address pools, you can duplicate some of the addresses among the pools. The address tracking mechanism of Steel-Belted Radius, when it is enabled, ensures that, if an IP address appears in more than one pool, after it is assigned out of any pool, it remains unavailable through any of the pools until it is released.

You must disable this type of address tracking if the server is assigning IP addresses from disjoint networks. In that configuration, two numerically identical IP addresses would signal a conflict, even though they actually belong to two different networks.

Order of Address Assignment
IP or IPX addresses are assigned on a first-in-first-out basis; that is, the address that was first released is the first to be reassigned. This ensures that addresses are out of use for as long as possible prior to reuse.

Concurrent Network Connections
The SBR Administrator program allows you to limit the number of active connections, on a per-user, or per-tunnel basis.

Concurrent User Connections
You can set a maximum limit on the number of concurrent connections that a user can have. Subsequently, when the user requests a new connection, Steel-Belted Radius compares the current number of connections to the maximum limit.

If a new connection would exceed the limit, Steel-Belted Radius can reject the additional connection or allow the connection, but log the event in the server log (described in “Using the Server Log File” on page 251).

NOTE: When counting connections, Steel-Belted Radius does not distinguish between multi-link connections and new user authentication attempts.

For concurrent connection limits to work, each NAD must be configured for RADIUS accounting and the same Steel-Belted Radius server must be responsible for both authentication and accounting. These conventions give the server full access to the data it needs to track connections accurately.

The maximum number of concurrent connections can be set individually for any type of user by selecting the Maximum Concurrent Connections check box and entering a number in the accompanying field in the appropriate Add User/Edit User dialog. See Chapter 5, “Administering Users” on page 47, especially “Concurrent Connection Limits” on page 65.

When a concurrent connection limit is set up for a user, it affects only that user. When a concurrent connection limit is set up for a group, every member of the group receives the same connection limit. For example, if GroupA has a connection limit of two, then each member of the group can have two concurrent connections.
Authentication methods that do not require user entries must provide alternate mechanisms for supporting concurrent connection limits. For example, if you are using external database authentication there is an alias mechanism you can use in the SQL or LDAP configuration file. Concurrent connection limits can be supported under proxy authentication only if the target server supports them.

**NOTE:** Concurrent user connections can be tracked across multiple Steel-Belted Radius servers by adding the Concurrency Server package.

### Concurrent Tunnel Connections

Steel-Belted Radius uses its Current Sessions list to determine the number of active connections for each tunnel. The Sessions list summarizes all of the RADIUS accounting data currently available to the server. Tunnel connections appear in the Sessions list using a special display convention that distinguishes them from user connections.

You can set a maximum limit on the number of concurrent connections that can be open using a specific tunnel. Subsequently, when a user requests a new connection through that tunnel, Steel-Belted Radius compares the current number of connections to the maximum limit. If a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.

For concurrent connection limits to work, it is essential that each NAD that can open a tunnel be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting. This permits the server’s Sessions list to be kept up to date and available to every NAD that needs to authenticate tunnel connections.

**NOTE:** Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Juniper Networks for more information.

### Phantom Records

When Steel-Belted Radius allocates resources such as IP addresses, IPX addresses, user connections, and tunnel connections, to its clients, it generates a *phantom* accounting record for its internal use. Phantom records are not written to the RADIUS accounting database, but they are displayed in the Current Sessions list (described on page 239). Phantom records resemble accounting start records, except that the session ID for phantom records is displayed as N/A.

After Steel-Belted Radius receives the corresponding accounting start request packet from the client, it discards the phantom record and replaces N/A with the actual Session-ID number returned by the client device in the Current Sessions list.

In some cases, Steel-Belted Radius can allocate a resource and create a phantom record, but then never receive a corresponding start packet from the client. To avoid committing the resource indefinitely, Steel-Belted Radius waits for a limited period for the start packet to confirm the transaction. By default, Steel-Belted Radius waits 180 seconds, though you can configure a different wait period by editing the radius.ini file (described in the Steel-Belted Radius Reference Guide).
Chapter 3  
Using SBR Administrator

This chapter presents an overview of how to use the SBR Administrator, which is a Java-based application that lets you configure settings for Steel-Belted Radius. In minutes, you can set up new users, alter standard profiles, or configure new network access devices from any computer on the network.

Running the SBR Administrator

You start the SBR Administrator by running a browser and opening a connection to the Steel-Belted Radius server you want to configure.

To log into a Steel-Belted Radius server:

1. Open a browser connection to the Steel-Belted Radius server you want to administer.
   - To administer a Steel-Belted Radius server running on your local host, enter http://localhost:port/, where port is the TCP port on which the server is listening for administration connections. For example, to open a connection on a local host listening on port 1812, use the following URL:

     http://localhost:1813/

   - To administer a Steel-Belted Radius server running on a remote host, enter http://server:port/, where server is the DNS name or IPv4/IPv6 address of the server, and port is the TCP port on which the server is listening for administration connections. For example, to open a connection on a remote host at IP address 192.168.24.15 listening on port 1812, use the following URL, where ipaddress is the IP address or DNS name of the remote server:


     When the Steel-Belted Radius Administrator page opens, click the Launch link to download and start the SBR Administrator.

2. When the Login dialog (Figure 8) opens, specify the TCP port you want to use for communication between SBR Administrator and the Steel-Belted Radius server.
The value you enter in the Port field must match the value specified in the SecureTcpAdminPort parameter in the [Ports] section of the radius.ini file on the Steel-Belted Radius server. The default value is 1813.

**Figure 8: Login Dialog**

![Login Dialog](image)

3. Enter your administrator username in the User Name field.

4. Enter your login password in the Password field.

5. Click Login.

When you click Login, SBR Administrator establish an HTTPS connection with the local or remote server. If it cannot establish a connection in 10 seconds, SBR Administrator times out and displays an error message.

---

**NOTE:** If a timeout occurs, verify that the Steel-Belted Radius service/daemon is running on the target server and that it is listening on the administration port you are specifying in the Login dialog.

SBR Administrator verifies that the username you entered is present in the access.ini file. If the username is found, SBR Administrator validates the password you entered against a local or remote password database.

When you connect to a server, the Status panel lists various features of the running server, such as version, platform on which it is running, IP address, available authentication methods, license information, and any initialization errors that might have occurred.

---

**Navigating in SBR Administrator**

This section describes how to use the SBR Administrator panels, menus, dialogs, and toolbar.

**SBR Administrator Panels**

SBR Administrator uses a series of framed windows (panels) to configure server settings and display statistics and logs in SBR Administrator. Figure 9 illustrates the components of an SBR Administrator panel. To display a panel, you click an entry in the Sidebar. SBR Administrator displays the specified panel in the Content frame.
Figure 9: SBR Administrator Panel Layout

**SBR Administrator Sidebar**

You use the entries in the sidebar in the SBR Administrator to select which panel you want to display; for example, to display the RADIUS Clients panel, you click **Radius Clients** in the sidebar.

Some entries in the sidebar are expandable. For example, when you click the **Users** entry in the sidebar, the entry displays subentries such as **Local** and **Domain**.

**SBR Administrator Menus**

The SBR Administrator has four menus: File, Edit, Web, and Help.

**File Menu**

Table 6 describes the functions of each entry in the File menu in the SBR Administrator.

<table>
<thead>
<tr>
<th>Table 6: File Menu Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Menu Entry</strong></td>
</tr>
<tr>
<td>License</td>
</tr>
<tr>
<td>Import</td>
</tr>
<tr>
<td>Export</td>
</tr>
</tbody>
</table>
Table 6: File Menu Options (continued)

<table>
<thead>
<tr>
<th>Menu Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>Opens the Login dialog, which lets you log in to a Steel-Belted Radius server. The Login dialog is described in “Running the SBR Administrator” on page 33. The Login menu entry is dimmed if you are logged into a Steel-Belted Radius server.</td>
</tr>
<tr>
<td>Logout</td>
<td>Terminates your connection to a Steel-Belted Radius server. The Logout menu entry is dimmed if you are not logged into a Steel-Belted Radius server.</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits the Steel-Belted Radius application.</td>
</tr>
</tbody>
</table>

Edit Menu

Table 7 describes the functions of each entry in the Edit menu in the SBR Administrator.

Table 7: Edit Menu Options

<table>
<thead>
<tr>
<th>Menu Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Deletes an existing object from the Steel-Belted Radius database and copies its information to the Clipboard. Active only when an object is selected in an SBR Admin panel.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the selected object from the Steel-Belted Radius database to the Clipboard. Active only when an object is selected in an SBR Admin panel.</td>
</tr>
<tr>
<td>Paste</td>
<td>Pastes an object from the Clipboard to the Steel-Belted Radius database. Active only after a Cut or Copy command has been used.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes an object from the Steel-Belted Radius database.</td>
</tr>
</tbody>
</table>

Web Menu

Table 8 describes the functions of each entry in the Web menu in the SBR Administrator.

Table 8: Web Menu Options

<table>
<thead>
<tr>
<th>Menu Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel-Belted Radius Home Page</td>
<td>Opens the support page for Steel-Belted Radius in a browser window. This page lets you review product information, download documentation and technical notes, and access other resources.</td>
</tr>
<tr>
<td>Juniper Networks Home Page</td>
<td>Opens the home page for Juniper Networks (<a href="http://www.juniper.net/">http://www.juniper.net/</a>) in a browser window.</td>
</tr>
</tbody>
</table>

Help Menu

Table 9 describes the functions of each entry in the Help menu in the SBR Administrator.

Table 9: Help Menu Options

<table>
<thead>
<tr>
<th>Menu Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Opens the online help for the SBR Administrator.</td>
</tr>
</tbody>
</table>
Navigating in SBR Administrator

Chapter 3: Using SBR Administrator

SBR Administrator Toolbar

After you log into Steel-Belted Radius, you can use the toolbar to manipulate SBR Administrator objects, such as users or RADIUS clients. The buttons on the SBR Administrator toolbar change when you change panels to provide buttons appropriate for your current context.

Figure 10: SBR Administrator Toolbar

Table 9: Help Menu Options (continued)

<table>
<thead>
<tr>
<th>Menu Entry</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>About</td>
<td>Displays the About SBR Administrator dialog, which lists version information for the SBR Administrator. For more information, see “Displaying Version Information” on page 40.</td>
</tr>
</tbody>
</table>

SBR Administrator Dialogs

This section summarizes how to use SBR Administrator dialogs and controls.

Adding an Entry

To add an entry to the Steel-Belted Radius database, open the appropriate panel and click the Add button on the SBR Administrator toolbar. The SBR Administrator displays an Add dialog (Figure 11).

Table 10: SBR Administrator Toolbar

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>Refreshes the displayed list of items in the SBR Administrator dialog.</td>
</tr>
<tr>
<td>Print</td>
<td>Prints the contents of the active panel.</td>
</tr>
<tr>
<td>Add</td>
<td>Adds an object to the Steel-Belted Radius database.</td>
</tr>
<tr>
<td>Edit</td>
<td>Edits an existing object in the Steel-Belted Radius database. Active only when an object is selected in the active panel.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes an existing object from the Steel-Belted Radius database.</td>
</tr>
<tr>
<td>Apply</td>
<td>In the Order of Methods dialog, applies any changes you have made to the authentication policy settings.</td>
</tr>
<tr>
<td>Reset</td>
<td>Restores the default values for controls in the active panel.</td>
</tr>
<tr>
<td>Clear (System Statistics only)</td>
<td>Resets statistics other than Server Up Time to zero.</td>
</tr>
<tr>
<td>Search (Filters only)</td>
<td>Opens the Search Filters window, which is described in “Searching the Filter List” on page 103.</td>
</tr>
<tr>
<td>EAP Setup (Order of Methods panel only)</td>
<td>Opens the EAP Setup dialog, which lets you specify the active EAP methods that will be used for an authentication method. For more information, see “Configuring EAP Settings” on page 149.</td>
</tr>
</tbody>
</table>
Every object of the same type must have a unique name. If the name you assign to an item is already being used by another item of the same type, the SBR Administrator displays a warning.

**Editing an Entry**

To edit an existing entry to the Steel-Belted Radius database, open the appropriate panel and double-click the item you want to change (or select the item and click the **Edit** button on the SBR Administrator toolbar). The SBR Administrator displays the settings for the item you selected in an Edit dialog. The **Save** button remains disabled until the contents of a field in the Edit dialog changes.

**NOTE:** You cannot change the name associated with an item in an Edit dialog. To change an item’s name, you must cut/paste the item and assign it the name you want it to have.

**Cutting/Copying/Pasting Records**

When you select an item from a panel displaying tables of items, you can choose **Edit > Cut** or **Edit > Copy** to cut or copy the item to the Clipboard, and then add a new record to the display by pasting it from the Clipboard by choosing **Edit > Paste**.

The Clipboard can contain one item of each type (RADIUS client, user, etc.) If you copy an item to the Clipboard and then copy another item of the same type, the information for the second item overwrites the information for the first item. Clipboard contents are preserved until you exit the SBR Administrator.

**Resizing Columns**

You can resize columns in an SBR Administrator table by dragging the column header boundary to the left or right.
**Changing Column Sequence**

You can change the sequence of columns in an SBR Administrator table by dragging the column headers left and right.

**Sorting Information**

By default, items in SBR Administrator tables are sorted by Name. You can sort items in any order by clicking a column header.

Previously sorted tables retain their order when the table is sorted on another column. If you want to sort a table by more than one column (for example, sort by address pool and subsort by IP address), click the least-significant column (here, IP Address), and then click the more significant columns (here, Address Pool).

---

**Adding License Keys**

Depending upon your purchasing arrangements, your Steel-Belted Radius software may require a new license key at some point after its initial installation.

If you are provided with a new license key by your reseller or by Steel-Belted Radius, you can add the key to an existing Steel-Belted Radius installation as follows:

1. Start the SBR Administrator program and connect to your Steel-Belted Radius server.
2. Select File > License.
3. When the Add a License for Server dialog (Figure 12) appears, enter the license key and click OK.

![Figure 12: Add a License for Server Dialog](image)

If the license key you have entered is invalid, the server displays an error message. If this occurs, click OK in the message dialog and enter the correct license key.

4. After you have entered a valid license key, the server displays a confirmation message and reminds you that you must restart the server. Click OK.

The server does not restart itself automatically after a new license key is added. You must restart Steel-Belted Radius manually to activate the new license key.
Accessing Online Help

To get help with the SBR Administrator, click the ? (help) button, press F1, or select Help > Contents.

Displaying Version Information

To identify the current version of the SBR Administrator, select Help > About to open the SBR Administrator dialog (Figure 13).

Figure 13: About SBR Administrator Dialog

Exiting the SBR Administrator

To exit the SBR Administrator, choose File > Exit.

Closing the SBR Administrator has no impact on the Steel-Belted Radius service or daemon.
Chapter 4

Administering RADIUS Clients

This chapter describes how to set up RADIUS clients and client groups.

A RADIUS client is a network device or software application that contacts Steel-Belted Radius when it needs to authenticate a user or to record accounting information about a network connection.

A RADIUS client group is a collection of network devices or software applications that contacts Steel-Belted Radius to authenticate a user or to record accounting information about a network connection. Members of a RADIUS client group use a contiguous range of IP addresses and use identical settings, such as a shared secret or an IP address pool.

RADIUS Clients Panel

The RADIUS Clients panel lets you identify the devices that you want to define as clients of Steel-Belted Radius.

Figure 14: RADIUS Clients Panel
Adding a RADIUS Client or Client Group

To add a RADIUS client or client group:

1. Choose RADIUS Clients in the sidebar.

2. Click the Add button.

The Add RADIUS Client dialog (Figure 15) appears.

Figure 15: Add RADIUS Client Dialog

3. Enter the name of the RADIUS client or client group in the Name field.

Although you can assign any name to a RADIUS client entry, you should use the device’s IP address or DNS hostname to avoid confusion.

You can create a special RADIUS client entry called <ANY> by clicking the Any RADIUS Client check box (Figure 16). The <ANY> RADIUS client allows Steel-Belted Radius to accept requests from any network access device or proxy RADIUS server, as long as the shared secret is correct.
Figure 16: Creating an *<ANY>* RADIUS Client

![Add RADIUS Client](image)

Note that the **IP Address** field for an *<ANY>* RADIUS client cannot be edited. *<ANY>* implies that the server accepts requests from any IP address, provided that the shared secret is correct.

4. Optionally, enter a description of the RADIUS client in the **Description** field.

   The description you associate with a RADIUS client is not used during processing.

5. Enter the IPv4 or IPv6 address of the RADIUS client in the **IP Address** field.

   You can enter the IPv4 or IPv6 address of the RADIUS client. Alternatively, you can enter the DNS name of the device; the SBR Administrator resolves the name you enter to its corresponding IP address and displays the result in the **IP Address** field.

   If you want the RADIUS client to use an IPv4 address range, enter the starting address for the range in the **IP Address** field, click the **Range** check box, and enter the number of addresses in the range in the **Range** field (Figure 17). You can create an address range of as many as 500 addresses in an address range.

   For more information on IPv4 address ranges for RADIUS clients, see “RADIUS Client Groups” on page 26.

Figure 17: Entering an IPv4 Address Range for a RADIUS Client

![Add RADIUS Client](image)

6. Enter the authentication shared secret for the RADIUS client in the **Shared secret** field.

   For privacy, asterisks are echoed as you type. You can check **Unmask** to display the characters in the shared secret.

   After you complete configuration of the authentication shared secret on the server side, you must enter the same authentication shared secret when you configure the network access device.

7. Use the **Make or model** list to select the make and model of your RADIUS client device.
The **Make or model** selection tells Steel-Belted Radius which dictionary of RADIUS attributes to use when communicating with this client. If you are not sure which make and model you are using or if your device is not in the list, select **Standard RADIUS**.

8. If you want the RADIUS client to obtain its IPv4 address from an address pool, click the **Address pool** check box and use the **Address pool** list to specify which address pool to use when authenticating an access request from this RADIUS client.

   Click the **View** button to display details for the address pool you select.

---

**NOTE:** You must configure IP address pools before you set up RADIUS clients if you want the clients to use address pools. For more information, see “Setting Up IP Address Pools” on page 85.

9. If you want to associate a profile with the RADIUS client, click the **Use Profile** check box and use the drop-down list to select the profile you want the RADIUS client to use.

   After you select a profile, you can click the **View** button to display the settings configured for the profile.

10. Specify how you want the profile to interact with the user settings.

    - If you want attributes in the profile to override identically-named attributes configured for the user, click the **Override User Attributes** check box.

    - If you want attributes in the profile to be merged with identically-named attributes configured for the user, clear the **Override User Attributes** check box and specify whether user or RADIUS client attributes should be used in the event they specify different values for the same attribute.

11. Optionally, specify an accounting secret for the RADIUS client.

   By default, Steel-Belted Radius uses the same shared secret for authentication and accounting. If you want the RADIUS client to use different shared secrets for authentication and accounting:

   a. Click the **Use different shared secret for accounting** check box.

   b. Click the **Edit** button.

   c. When the Accounting Shared Secret dialog (Figure 18) opens, enter the shared secret you want the RADIUS client to use for accounting.

   *Figure 18: Accounting Shared Secret Dialog*
For privacy, asterisks are echoed as you type. You can click the Unmask check box to display the characters in the shared secret.

d. Click OK.

You must enter the same accounting shared secret when you configure the RADIUS client.

12. Optionally, indicate whether you want to enable keepalive processing and specify how long the server should wait for RADIUS packets from the client before assuming connectivity has been lost.

If you check the Assume down if no keepalive packets after check box, you can enter a value in the (seconds) field. If the server does not receive any RADIUS packets from this client after the specified number of seconds, the server assumes that the connection to the client is lost or that the client device has failed. When this happens, Steel-Belted Radius gracefully closes any user or tunnel connections it has authenticated for the client. Steel-Belted Radius releases any pooled IP or IPX addresses and adjusts the counts of concurrent user or tunnel connections appropriately.

NOTE: If the value you enter in the (seconds) field is too low, valid user or tunnel connections can be lost. For example, during low usage periods, a network access device might send no RADIUS packets to the Steel-Belted Radius server, even though the device is still “up.”

13. When you are finished, click OK.

Verifying a Shared Secret

To verify a shared secret on Steel-Belted Radius:

1. Open the RADIUS Clients panel.

2. Select the RADIUS client whose shared secret you want to verify and click the Edit button (or double-click the RADIUS client entry).

   The Edit RADIUS Client dialog opens.

3. Enter the shared secret you think is assigned to the RADIUS client in the Shared secret field.

4. Click the Validate button.

   The SBR Administrator displays a message indicating whether you entered the correct shared secret.
Deleting a RADIUS Client

To delete a RADIUS client:

1. Open the RADIUS Clients panel.
2. Select the RADIUS client entry you want to delete.
3. Click the **Delete** button on the SBR Administrator toolbar.
4. When you are prompted to confirm the deletion request, click **Yes**.
Chapter 5
Administering Users

This chapter describes how to add users to the Steel-Belted Radius database.

User Files

The following files establish settings for setting up users. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

**Table 11: User Account Files**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>redirect.ini</td>
<td>Configures settings for when Steel-Belted Radius should redirect users after repeated failed login attempts.</td>
</tr>
<tr>
<td>radius.ini</td>
<td>Specifies (among other things) the settings relating to RSA SecurID support in Steel-Belted Radius.</td>
</tr>
<tr>
<td>securid.ini</td>
<td>Specifies the prompt strings returned to SecurID users during login and authentication.</td>
</tr>
<tr>
<td>tacplus.ini</td>
<td>Specifies the name of the TACACS+ server and the shared secret used to validate communication between the Steel-Belted Radius server and the TACACS+ server.</td>
</tr>
</tbody>
</table>

Users Panels

The Users entry in the sidebar has as many as five sub-entries, as described in Table 12. Each user entry in the Steel-Belted Radius database identifies one method by which the server can authenticate a specific user.

**Table 12: User Panels**

<table>
<thead>
<tr>
<th>User Panel</th>
<th>Function</th>
<th>Available On Server Running OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>Lists the native users in the local Steel-Belted Radius database. You must display the Native User panel to add, edit, and delete native users.</td>
<td>Windows, Linux, Solaris</td>
</tr>
<tr>
<td>Domain</td>
<td>Lists the users authenticated using Windows Domain authentication. You must display the Domain tab to add, edit, and delete Domain users.</td>
<td>Windows</td>
</tr>
</tbody>
</table>
Setting Up Native Users

Native user entries require you to enter the user’s name and password into the Steel-Belted Radius database. For all other types of user entry, the server relies on another database to confirm the user’s password.

**NOTE:** You must define a native user entry for every user who requires remote access to a Windows network. For example, you can accommodate Solaris, Linux, or Macintosh users by adding them as native users.

**Adding a Native User**

To add a native user to the Steel-Belted Radius database:

1. Choose Users > Native in the sidebar.

The Native Users panel (Figure 19) appears.

---

<table>
<thead>
<tr>
<th>User Panel</th>
<th>Function</th>
<th>Available On Server Running OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurID</td>
<td>Lists the users authenticated using RSA SecurID authentication. You must display the <strong>SecurID</strong> tab to add, edit, and delete SecurID users.</td>
<td>Windows Linux Solaris</td>
</tr>
<tr>
<td>TACACS+</td>
<td>Lists the users authenticated using TACACS+. You must display the <strong>TACACS</strong> + tab to add, edit, and delete TACACS+ users.</td>
<td>Windows Linux Solaris</td>
</tr>
<tr>
<td>UNIX</td>
<td>Lists the users and groups authenticated using UNIX authentication. You must display the <strong>UNIX</strong> tab to add, edit, and delete UNIX users.</td>
<td>Linux Solaris</td>
</tr>
</tbody>
</table>

---

**NOTE:** You can populate the user database for Steel-Belted Radius by entering information in the Users panel or by importing data from other servers. For more information on importing user information, see Appendix E, “Importing and Exporting Data.”
2. Click Add.

The Add Native User dialog (Figure 20) appears.

3. Enter the user’s login name in the Name field.

Native user entries in the Steel-Belted Radius database have all-uppercase names; names are converted to all-uppercase letters when the native user entry is created, and they remain all-uppercase for the life of the entry. For example, a native username entered as realLife1 is stored as REALLIFE1 in the Steel-Belted Radius database.
4. Optionally, enter a description of the user in the **Description** field.

The description you associate with a native user is not used during processing.

5. Enter the user’s login password in the **Password** field.

If you want the characters in the password (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that passwords are case-sensitive: `swordfish`, `SwordFish`, and `SWORDFISH` are three different passwords.

6. Specify whether you want the user’s password to be encrypted before it is stored.

   - If this user requires only PAP authentication and you want to store the hash of the password in the Steel-Belted Radius database, click the **Store hash of password** check box. This option allows the user to authenticate using only PAP.
   
   - If this user requires CHAP authentication, do not click the **Store hash of password** check box.

7. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

For more information on profiles, refer to Chapter 6, “Administering Profiles.”

8. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

Refer to “Adding a Checklist or Return List Attribute for a User” on page 52 for information on how to add checklist and return list attributes.

9. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

10. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

11. Click **OK**.

**Editing a Native User**

After you have added a user, you can modify any setting for that user except the username. To edit a native user who already exists in the Steel-Belted Radius database:

1. Choose **Users > Native** in the sidebar.

The Native Users panel (Figure 19 on page 49) appears.
2. Select the user entry you want to edit and click the **Edit** button (or right-click an entry and choose **Edit** from the context menu).

The Edit Native User dialog (Figure 21) opens.

**Figure 21: Edit Native User Dialog**

![Edit Native User Dialog](image)

3. Edit the settings for the user as appropriate.

Refer to “Adding a Native User” on page 48 for information on the fields in the native user dialogs.

You can modify any setting except the user’s name. To edit a user’s name, you must copy the user record to a new user entry.

4. Click **Save**.

**Deleting a Native User**

To delete a native user:

1. Choose **Users > Native** in the sidebar.

   The Native Users panel (Figure 19) appears.

2. Select the user entry you want to delete and click the **Delete** button (or right-click an entry and choose **Delete** from the context menu).

3. When you are asked to confirm the deletion, click **Yes**.
**Adding a Checklist or Return List Attribute for a User**

A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated.

A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

To add a checklist or return list attribute to a user’s entry:

1. Open the appropriate user entry.
2. Click the **Checklist** tab or the **Return list** tab.
3. Click **Add**. The Add Checklist Attribute dialog or the Add Return List Attribute dialog (Figure 22) opens.

**Figure 22: Add Checklist Attribute and Add Return List Attribute Dialogs**

4. Select the attribute you want to add from the **Attributes** list.
5. Select or enter a value for the attribute.

The dialog changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.

If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

(Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

(Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.
6. Click **Add** to add this attribute/value pair to the list.

7. When you are finished adding attribute/value pairs, click **Close** to return to the Add User dialog.

---

**Setting Up Windows Domain Users**

To use the Windows Domain Authentication plug-in, the RADIUS service must be run under the **LocalSystem** account on a Windows XP computer that is part of a domain. (The **LocalSystem** account is a standard authenticated domain user account in Windows.) Groups and users to be authenticated can reside in any domain within the forest, as well as in those domains outside the forest for which a trust relationship exists.

When you use Windows domain authentication, the domain name can be present in the User-Name attribute of the Access Request, which can be of the form `\domain\user`, `domain\user`, or simply `user`. Additionally, the form `user@domain` can be used.

**Prequalification Checklists**

By default, when Steel-Belted Radius uses Windows domain membership to authenticate a user, it processes attributes for the first group that the user matches. The attributes consist of checklist and reply-list attributes, and checklist processing is performed to determine the user’s authorization rights after authentication succeeds.

If an enterprise sets up separate Windows domain groups for different access methods (for example, one domain group for users accessing the network through a VPN and another domain group for users accessing the network through a WLAN Access Point) and then assigns users to more than one domain group (so that the users get different permissions based on what access method they use), Steel-Belted Radius can authenticate the user against the first group the user matches and process the wrong attributes for that user, causing checklist processing to fail and the user’s access to be rejected.

Prequalification checklists allow a site to perform checklist processing before it authenticates a user, so that the attributes returned by every group a user belongs to can be evaluated (and the appropriate membership chosen) before authentication proceeds.

Example: CandyCorp sets up two groups (WLAN_USERS and VPN_USERS) in the CORP domain and creates access policies for each. Mary is a member of both groups; when she accesses the corporate network through a WLAN Access Point, her traffic should be tagged for a specific VLAN, and when she accesses the corporate network through a VPN, an Ascend-Data-Filter should be sent to the VPN gateway to restrict the internal hosts she can reach.
Without prequalification checklist processing, Steel-Belted Radius responds to Mary’s connection through an Access Point by using the first domain group membership it finds (which might be VPN_USERS), authenticating Mary and returning the attributes associated with that group, and then rejecting Mary because post-authentication checklist processing fails when the group used for authentication (VPN_USERS) didn’t provide the appropriate access attributes.

With prequalification checklist processing enabled, Steel-Belted Radius responds to Mary’s connection through the Access Point by running checklist processing before it authenticates Mary. Steel-Belted Radius tests each group to which Mary belongs to see if authentication and authorization will ultimately be successful. If checklist processing for a domain group fails, that group is skipped and the next group is tried; if checklist processing for all groups fails, Mary’s access request is denied. If checklist processing successfully matches Mary to a domain group, authentication proceeds, and Mary’s traffic is processed according to corporate policies (that is, it is tagged with the VLAN identifier appropriate for her WLAN access).

The application of prequalification checklist processing is not limited to domain groups. Prequalification checklists can be used to direct a user request to an appropriate domain user entry based on the presence of attributes in the user’s request. For example, if a user’s name ("ADMIN") is specified in an Access-Request and both \CORP\ADMIN and \LAB\ADMIN are listed in the Steel-Belted Radius database with the same password, prequalification checklist processing could be used to select the appropriate domain user object for authentication and authorization.

Prequalification processing is enabled through the PrequalifyChecklist argument in the [Windows Domain] section of the winauth.aut file.

**MS-CHAP Considerations**

If the user is successfully authenticated, any appropriate encryption keys (obtained through either MS-CHAP or MS-CHAP-V2) are returned to Steel-Belted Radius and the user’s profile is retrieved from the Steel-Belted Radius database. To enable encryption, the appropriate attributes, such as Mppe-Send-Key and Mppe-Recv-Key, must be included in the user’s profile. You do not need to prepend the username with the domain name to avoid timeout problems when dealing with a large number of domains.

The Windows Domain Authentication plug-in does not support EAP pre-fetch.


**Expired Domain Passwords**

The Windows domain authentication method allows users to be authenticated against domain security using an expired domain password. This lets Steel-Belted Radius handle security policies that force domain passwords to be changed automatically after a certain number of days. Typically, after the password expires, at the next attempt to log in, the domain recognizes the password supplied by the user as expired. The domain then returns a special status code to its client application indicating these conditions. Typically, the user is then prompted to change his or her domain password, but the client application (for example, Microsoft Remote Access Client) must support the ability to change passwords.

When Steel-Belted Radius passes a username/password pair through to a domain for authentication, the domain can indicate to Steel-Belted Radius that the password is expired. If so, Steel-Belted Radius’s default response is to issue an Access-Reject. You can configure it to respond instead with an Access-Accept.

**Windows Domain Authentication Configuration**

As with other authentication plug-ins, winauth.dll is configured through a single .aut file (winauth.aut). The winauth.aut file must contain [Bootstrap] and [Windows Domain] sections:

```plaintext
[Bootstrap]
LibraryName=winauth.dll
Enable=1
InitializationString=Windows domain authentication
```

Processing for users with expired passwords is configured in the [Windows Domain] section:

```plaintext
[Windows Domain]
AllowExpiredPasswordsForUsers = no
AllowExpiredPasswordsForGroups = no
RetryFailedAuthentications = no
AllowMachineLogin = yes
;ProfileForExpiredUsers = Profile
;ProfileForExpiredUsersInGroups = Profile
PrequalifyChecklist = no
```

**NOTE:** MS-CHAP and MS-CHAP-V2 users with expired passwords are not accepted. They may be prompted to change password if their login application supports password changing.

**Adding a Domain User or Domain Group**

To use domain authentication, the Steel-Belted Radius service must run on a Windows workstation or server that belongs to a domain. The Windows host running Steel-Belted Radius does not need to be a domain controller.

It is possible to authenticate against domains other than the one in which the Steel-Belted Radius service is running, provided that the other domain is trusted by the domain of the RADIUS service. The trust relationship may not be mutual; the other domain does not have to trust the RADIUS domain.
Example: An enterprise has three domains: A, B, and C, and Steel-Belted Radius is running in A. A trusts B and C trusts A. You can use Domains A and B for authentication, but not C, because A does not trust C.

You can add a Domain User entry to provide for the authentication of a specific user defined within a specific domain under Microsoft networking. For more flexibility, you can add a Domain Group, to provide for the authentication of all users that belong to a specific group defined within a specific domain.

To add a domain user or domain group:

1. Choose Users > Domain in the sidebar.

2. Click the Add button on the SBR Administrator toolbar to display the Add Domain User dialog (Figure 23).

3. Specify the domain and username for the user you want to add.

   You can enter the user’s domain in the Domain field and the user’s login name in the User field.

   Domain usernames must be in the format `\domain\user`. Domain usernames cannot contain the following characters:

   `\ / [ ] : | < > = ; , * ? @`

   If you want to browse for an existing user or group, click the Browse button.

   When the Browse for Domain User dialog (Figure 24) opens, click the name of the appropriate domain, and then click the name of the user or group in that domain you want to use. Click OK to finish.
4. Optionally, enter a description of the domain user or group in the **Description** field.

The description you associate with a native user is not used during processing.

5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

   For more information on profiles, refer to Chapter 6, “Administering Profiles.”

6. If you want to specify checklist attributes or return list attributes for the domain user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to “Adding a Checklist or Return List Attribute for a User” on page 52 for information on how to add checklist and return list attributes.

7. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

8. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

9. Click **OK**.
Setting Up SecurID Users

Usernames are case-sensitive. The case in which names are recorded depends on whether usernames are being stored in a local or external database. Usernames stored in an external database (UNIX, RSA SecurID, TACACS+) retain their case as stored in that database.

Adding a SecurID User

You can configure Steel-Belted Radius to use RSA SecurID authentication for your users by setting up communication between the RSA server and the RADIUS server (described in “Configuring SecurID Authentication” on page 147), and then adding SecurID users to the Steel-Belted Radius database using the instructions that follow.

Steel-Belted Radius attempts SecurID authentication only on usernames that match a SecurID entry in its User database. Steel-Belted Radius offers four types of SecurID entry, each providing a different matching rule:

- You can enter the name of a specific user.
  
  For example, you might create a SecurID user entry for the specific user George. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for username George. If username George is authenticated, the attributes of the user entry apply.

- You can enter a prefix.
  
  For example, you might create a SecurID entry for the prefix sales$. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for a username such as sales$Harry or sales$Cynthia. Using a prefix lets you group multiple SecurID users into a single user entry instead of creating a separate entry for each SecurID user. If the sales$ user is authenticated, the attributes of the user entry apply.

- You can enter a suffix.
  
  A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

- You can create an entry for Any user.

NOTE: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the RSA SecurID server.

You can use different settings for different groups.

NOTE: The user must type in the prefix as part of the username when dialing in and requesting a connection.

- You can enter a suffix.

A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

- You can create an entry for Any user.
This creates a single user entry named <ANY> that matches any username to be authenticated. SecurID can be used as an authentication method for any username, and, if successful, the attributes of the <ANY> entry apply.

The <ANY> entry makes sense if a single set of attributes apply to all your SecurID users and if you want to make SecurID either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a SecurID user entry:

1. Choose Users > SecurID in the sidebar.
2. Click the Add button on the SBR Administrator toolbar to display the Add SecurID User dialog (Figure 25).

   **Figure 25: Add SecurID User Dialog**

3. Enter the specific username, a prefix, or a suffix in the Name field.
4. Select the user type: Specific user, Prefix, Suffix, or Any user.
5. Optionally, enter a description of the user in the Description field.
6. If you want to use a profile to assign checklist and return list attributes to the user, click the Use profile check box and use the Profile list to select the profile you want.

   After you select a profile, you can click the View button to display the checklist and return list attributes in that profile.

7. If you want to specify checklist attributes or return list attributes for the user, click the Checklist tab or the Return list tab, and then click the Add button.
Refer to “Adding a Checklist or Return List Attribute for a User” on page 52 for information on how to add checklist and return list attributes.

8. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

9. If you want to specify the maximum number of concurrent connections this user can maintain, click the Maximum concurrent connections check box and enter a number in the accompanying field.

10. Click OK.

Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string USERNAME; for example, !USERNAME or SALES<USERNAME>.

---

**Setting Up TACACS+ Users**

You can configure Steel-Belted Radius to authenticate your users by querying a TACACS+ server.

**NOTE:** Before you add TACACS+ users, you must configure communication between the Steel-Belted Radius server and the TACACS+ server by editing the tacplus.ini file. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

Steel-Belted Radius attempts TACACS+ authentication only on usernames that match a TACACS+ entry in its user database. Each type of TACACS+ entry specifies a different matching rule:

- You can enter the name of a specific user.

For example, you might create a TACACS+ user entry for the specific user George. This tells Steel-Belted Radius that when an authentication request is received for username George, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

- You can enter a prefix.

For example, you might create a TACACS+ user entry for the prefix sales$. This tells Steel-Belted Radius that when an authentication request is received for a username such as sales$Harry or sales$Cynthia, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

**NOTE:** Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the TACACS+ server.
Using a prefix lets you group multiple TACACS+ into a single user entry instead of creating a separate entry for each TACACS+ user. You can use different settings for different groups.

**NOTE:** The user must type in the prefix as part of the username he or she is using to dial in and request a connection.

- You can enter a suffix.

A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were `!sales`, you might have usernames such as `Harry!sales` or `Cynthia!sales`.

- You can create an entry for *Any* user.

This creates a single user entry named `<ANY>` that matches any username to be authenticated. TACACS+ can be used as an authentication method for any username, and, if successful, the attributes of the `<ANY>` entry apply.

The `<ANY>` entry makes sense if a single set of attributes apply to all your TACACS+ users and if you want to make TACACS+ either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a TACACS+ user:

1. Choose **Users > TACACS+** in the sidebar.
2. Click the **Add** button on the SBR Administrator toolbar to display the Add TACACS+ User dialog (Figure 26).

**Figure 26: Add TACACS+ User Dialog**
3. Enter the specific username, a prefix, or a suffix in the **Name** field.

4. Select the user type: **Specific user, Prefix, Suffix**, or **Any user**.

5. Optionally, enter a description of the user in the **Description** field.

6. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

7. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to “Adding a Checklist or Return List Attribute for a User” on page 52 for information on how to add checklist and return list attributes.

8. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

9. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

10. Click **OK**.

   Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string **USERNAME**; for example, !USERNAME or SALES<USERNAME>.

---

**Setting Up UNIX Users**

You can add a UNIX user entry to provide for the authentication of a specific user defined on a Linux or Solaris server. For more flexibility, you can add a UNIX group to provide for the authentication of all users that belong to a specific group defined on the server.

To add a UNIX user or group:

1. Choose **Users > UNIX** in the sidebar.

2. Click the **Add** button on the SBR Administrator toolbar to display the Add UNIX User dialog (Figure 27).
3. Click the **Browse** button and select a user or group from the list. Click **OK**.

4. Optionally, enter a description of the user in the **Description** field.

5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

6. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to “Adding a Checklist or Return List Attribute for a User” on page 52 for information on how to add checklist and return list attributes.

7. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

8. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

9. Click **OK**.
Editing User Settings

This section describes fields that you can set for any user entry, regardless of user type. For more information, see “User Attribute Lists” on page 11 and “About Profiles” on page 67.

Selecting a Profile

To select a profile for a user:

1. Open the appropriate user panel by clicking a User > entry in the sidebar.
2. Select the user whose entry you want to modify.
3. Click Edit (or double-click the user entry).
4. Click the Use Profile check box.
5. Select the list to select the profile you want to use.

   To display the settings associated with the selected profile, click the View button.

6. When you are finished, click Save.

Setting Attribute Values

To change the value of an attribute already in the checklist or return list for a user entry:

1. Click the Checklist tab or Return List tab.
2. Select the attribute whose value you want to change.
3. Click Edit or double-click the attribute.
4. When the Change dialog opens, enter or select the new value.

   Depending on the attribute, you can enter a new value or select a value from a list. For some attributes, Steel-Belted Radius retrieves the value from the server and you cannot enter a value in this dialog.

5. Click OK.
**Removing Attribute/Value Pairs**

To remove an attribute/value pair already in the checklist or return list for a User entry:

1. Click the **Checklist** tab or **Return List** tab.
2. Select the attribute/value pair you’d like to remove.
3. Click **Delete**.

**Reordering Attributes**

Certain attributes are multi-valued and orderable; that is, the attribute/value pair can appear more than once in a RADIUS response, and the order in which the attribute/value pairs appear is significant.

To reorder attributes in a User entry:

1. Click the **Checklist** tab or **Return List** tab.
2. Highlight an attribute/value pair in the list.
3. Click the Up or Down arrow to move the selected attribute within the list.
   - The Up arrow moves the selected attribute/value up in the list. If the attribute is not orderable, or if the selected item already the first value for this attribute, the button is disabled.
   - The Down arrow moves the selected attribute/value down in the list. If the attribute is not orderable, or if the selected item is already the last value for this attribute, the button is disabled.

**Changing Attributes Inherited from a Profile**

Checklist and return list attributes can be specified for a user, or they can be inherited from a profile associated with a user. Attributes inherited from a profile are overridden by attributes assigned to a specific user.

**Concurrent Connection Limits**

A maximum number of open connections can be set for each user entry by checking the **Maximum concurrent connections** check box and entering a number in the accompanying field. When the user requests access, the user can be authenticated using the given authentication method only if fewer than this number of connections are currently open for this user.

---

**Deleting a User**

To delete a user:

1. Open the appropriate user panel by clicking a User > entry in the sidebar.
2. Select the user you want to delete.

3. Click the **Delete** button from the SBR Administrator toolbar (or right-click the user entry and choose **Delete** from the context menu).

4. When you are prompted to confirm the deletion, click **Yes**.
Chapter 6

Administering Profiles

This chapter describes how to set up and administer user profiles.

About Profiles

Steel-Belted Radius lets you define default templates of checklist and return list pairs called profiles. A profile provides specific attributes for one or both lists. You can define as many profiles as you require. Profiles provide a powerful means of managing and configuring accounts.

When you edit a User entry, you can assign a profile to the User; the checklist and return list attributes of that profile then become the default settings for the User entry. After you assign a profile to a User entry, you can modify the new entries on the user’s checklist and return list. Changes you make apply only to the specific user entry; they do not affect the profile itself. Assigning a profile and then overriding individual attributes is a convenient way to leverage Steel-Belted Radius’s features to your advantage.

To change attributes settings across many users immediately, edit the profile that you have assigned to these users. The changes you make to a profile are automatically reflected in each user’s checklist and return list.

Adding a Checklist or Return List Attribute for a Profile

A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated. A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

Resolving Profile and User Attributes

If user-specific attributes are stored in an external database, Steel-Belted Radius determines the final set of attributes for a user by merging the attributes stored in the native database with those retrieved from the external database. This calculation is performed as follows:

1. The attributes from the profile (or Alias user) assigned to the user are first retrieved.
2. These attributes are then merged with the user-specific modifications to the attributes in the following manner:

- If the attribute is multi-valued, then the attribute(s) retrieved from the external database is added to the overall list of attributes.

- If the attribute is single-valued, then the attribute(s) retrieved from the external database replaces any attribute of the same name in the profile or associated with the alias.

- If the attribute is orderable, then the attribute(s) retrieved from the external database replaces any orderable attribute of the same name in the profile or associated with the alias.

## Setting Up Profiles

The Profiles panel (Figure 28) lets you define sets of checklist and return list attributes. You can then assign these profiles to users to simplify user administration.

### Figure 28: Profiles Panel

Adding a Profile

To add a profile:

1. Click **Profiles** to open the Profiles panel.

2. Click the **Add** button on the SBR Administrator toolbar.

   The Add Profile dialog (Figure 29) appears.
3. Enter a name for the new profile in the Name field.

4. Optionally, enter a description for the profile in the Description field.

5. Add checklist and return list attributes to the profile.
   a. Click the Checklist tab or the Return list tab.
   b. Click Add. The Add Checklist Attribute dialog or the Add Return List Attribute dialog (Figure 30) opens.

   Figure 30: Add Checklist Attribute and Add Return List Attribute Dialogs

   c. Select the attribute you want to add from the Attributes list.
   d. Select or enter a value for the attribute.

   The dialog changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.
If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

(Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

(Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.

e. Click **Add** to add this attribute/value pair to the list.

f. When you are finished adding attribute/value pairs, click **Close** to return to the Add Profile dialog.

6. Click **OK** to save the profile.

### Removing a Profile

To remove a profile:

1. Open the Profiles panel.

2. Select the entry for the profile you want to remove.

3. Click the **Delete** button on the SBR Administrator toolbar (or right-click the profile entry and choose **Delete** from the context menu).

4. When you are prompted to confirm the deletion, click **Yes**.

---

**NOTE:** Do not delete a profile that is assigned to a user. If you delete an active profile, the attributes defined in the profile are removed from users settings, possibly resulting in authentication failures.
Chapter 7
Administering Proxy RADIUS

This chapter presents an overview of proxy RADIUS and describes how to set up proxy targets.

About Proxy RADIUS

Steel-Belted Radius can forward a RADIUS request to another server for processing and relay the other server's result back to its client. Steel-Belted Radius is acting as a proxy for the target server, and that Steel-Belted Radius is proxy-forwarding the request to the target server.

Any Steel-Belted Radius server can act as proxy or target for authentication or accounting messages (or both).

Proxy RADIUS Authentication

Figure 31 illustrates how RADIUS authentication messages are proxy-forwarded:

1. A network access device (RADIUS client) sends an authentication request to a RADIUS proxy server.
2. The proxy RADIUS server forwards the message to a RADIUS target server.
3. The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.
4. The proxy RADIUS server relays the response message to the RADIUS client.

Figure 31: RADIUS Proxy Forwarding
Proxy RADIUS Accounting

RADIUS accounting messages are proxy-forwarded as follows:

1. A RADIUS server receives an accounting request.

2. Depending on its configuration, the RADIUS server forwards the accounting message to a target server, records accounting attributes locally on the proxy server, or records the information in both places.

3. If the proxy server does not receive an acknowledgement of the forwarded packet, it periodically re-sends the packet according to its retry policy.

Adding a Proxy Target

This section explains how to set up proxy forwarding from the Steel-Belted Radius server (the proxy) to another RADIUS server (the target).

To add a proxy target:

1. Open the Proxy Target panel.

2. Click the Add button on the SBR Administrator toolbar.

The Add Proxy Target dialog (Figure 32) appears.

Figure 32: Add Proxy Target Dialog

3. Enter the name of the proxy target in the Name field.

The target name must not duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration. The name you record for a proxy target is not used in processing; Steel-Belted Radius uses the proxy target’s IP address to route RADIUS packets.
4. Enter a description for the proxy target in the **Description** field.

5. Enter the IP address or DNS name of the proxy target in the **IP Address** field.

   If you enter the DNS name of the proxy target, the SBR Administrator resolves the name you enter to an IP address automatically.

6. Enter the shared secret for the proxy target in the **Shared Secret** field.

   Note that shared secrets are case-sensitive.

   If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box.

   The shared secret configured for the proxy target in Steel-Belted Radius must match the shared secret configured on the proxy target.

7. Specify how many times Steel-Belted Radius should try to reach the proxy target and how long to wait between attempts in the **Number of retries** and **Milliseconds between retries** fields.

   When Steel-Belted Radius acts as a proxy, it emulates the characteristics of a network access device. This includes the ability to retransmit a request if the first attempt does not get a timely response from the proxy target.

   - The **Number of retries** field specifies the number of times a request is retransmitted if an acknowledgment from the target is not received; if the number of retries is exhausted, then the original request is rejected. By default, Steel-Belted Radius retries three times before giving up.

   - The **Milliseconds between retries** field specifies the time interval between each retry in milliseconds (thousandths of a second). By default, Steel-Belted Radius waits 5000 milliseconds (5 seconds) between retries.

8. If the proxy target uses ports different from what the proxy intermediary uses for authentication or accounting, click the **Authentication** or **Accounting** check box and enter the port number you want Steel-Belted Radius to use when exchanging RADIUS authentication or accounting information with the proxy target.

   The port numbers configured for the proxy target in Steel-Belted Radius must match the port numbers configured on the proxy target. By default, Steel-Belted Radius uses port 1645 for authentication and port 1646 for accounting.

9. Specify whether you want accounting requests to be forwarded or recorded locally.

   - If you click the **Forward** check box, Steel-Belted Radius forwards the accounting transaction to the same proxy target that received the authentication transaction.

   - If you click the **Record locally** check box, Steel-Belted Radius logs the accounting transaction locally (regardless of whether an authentication request was forwarded to the proxy target).
You can click both check boxes if you want accounting requests to be forwarded and logged locally.

10. If you want Steel-Belted Radius to use a different shared secret for accounting when communicating with the proxy target, click the **Use different shared secret for accounting** check box and click the **Edit** button to specify an accounting shared secret.

Refer to “Maintaining an Accounting Shared Secret” on page 74 for information on using the Edit Accounting Shared Secret dialog.

11. If you want to use a proxy target as an authentication method, click the **Make available as an authentication method** check box.

If you enable this option, the name of the proxy target appears in the Authentication Methods tab of the Authentication Policies panel as `proxy:name`. This is useful if you have user records defined on an older RADIUS server and you want to provide a seamless migration to Steel-Belted Radius. Using the older server as a proxy RADIUS target means that RADIUS requests that arrive addressed to this target are handled by Steel-Belted Radius automatically, without requiring end users to change their addressing conventions.

12. Click **OK**.

**NOTE:** If the proxy target that you are configuring is a member of a proxy RADIUS realm, you should ensure that the **Make available as an authentication method** check box is unchecked.

Ask the administrator at the target site to log into the target server’s RADIUS configuration program and add Steel-Belted Radius as a RADIUS client of the target server. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

### Maintaining an Accounting Shared Secret

To specify a shared secret for accounting:

1. Click the **Use different shared secret for accounting** check box.

2. Click the **Edit** button.

3. When the Accounting Shared Secret dialog (Figure 33) opens, enter the shared secret you want Steel-Belted Radius to use.

**Figure 33: Accounting Shared Secret Dialog**
If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that shared secrets are case-sensitive.

4. Click **OK**.

---

## Deleting a Proxy Target

To remove a target server from the proxy target list:

1. In the Proxy Target panel, select the target server you’d like to remove.

2. Click **Delete**.

---

## Steel-Belted Radius as a Target

This section describes how to set up proxy forwarding from some other RADIUS server (the proxy) to the Steel-Belted Radius server (the target):

1. Set up the proxy as a RADIUS client of Steel-Belted Radius.

   Add the entry using the RADIUS Clients panel. Specify the proxy’s name, its IP address, and the shared secret that you want to use for encryption between the proxy and Steel-Belted Radius.

2. Ask the administrator at the target site to log into the proxy’s RADIUS configuration program and set up Steel-Belted Radius as a proxy RADIUS target. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

---

**NOTE:** Make sure that the same UDP port and shared secret are entered on both proxy and target sides.

---

### Dictionaries when Steel-Belted Radius is the Target

When Steel-Belted Radius receives a proxy-forwarded packet, it consults its RADIUS client entry for that proxy server. The **Make/model** field of this entry determines which attribute dictionary Steel-Belted Radius uses.

At various different times, Steel-Belted Radius can receive requests from the same proxy server that have originated from different network access devices, possibly of different types. The single **Make/model** field that was entered for the proxy might not be adequate to handle the variety of RASs on the “other side” of the transaction.
One way to handle this problem is to add the originating network access devices to Steel-Belted Radius’s list of RADIUS clients. Steel-Belted Radius can be configured to examine each proxy-forwarded packet for clues as to the make and model of the originating device. If clues are found, Steel-Belted Radius does everything it can to map this information to a vendor-specific dictionary, and uses this dictionary in preference to the one for the proxy.

**Accepting Packets from Any Proxy**

If you’d like Steel-Belted Radius to be able to accept proxy requests from any IP address, you can use the RADIUS Clients panel to add a special entry called <ANY>, and specify a shared secret. The <ANY> entry permits forwarded requests from any proxy to be accepted, provided the shared secret is correct.

**NOTE:** This feature requires that proxies are configured to use the shared secret you provide in the <ANY> entry.

**Proxy RADIUS as an Authentication Method**

Any target proxy RADIUS server can be configured as a Steel-Belted Radius authentication method by enabling the Make available as an authentication method check box in the Add Proxy Target/Edit Proxy Target dialog.

A target server can be set up as an authentication method even if the end users do not know anything about the target. That is, a user does not need to log in using a decorated username such as User@TargetName to be authenticated by the target server.

If you prioritize the proxy: TargetName authentication method above the Native User authentication method in the authentication methods list, the user can log in as User and Steel-Belted Radius automatically sends the request to the target for authentication. The authentication succeeds if the UserName and password are stored on the target, but if not, Steel-Belted Radius reaches the Native User method eventually, and the user can then be authenticated.

This technique is useful as a migration path to Steel-Belted Radius from other RADIUS servers. You can set up Steel-Belted Radius as the proxy and the old RADIUS server as the target. After proxy authentication is enabled (in the Proxy Targets panel) and prioritized (in the Authentication Policies panel), Steel-Belted Radius can authenticate users against the old RADIUS server, either as an automatic “first choice” or as an alternative when authentication against the new server’s “local” database fails.
Chapter 8
Administering RADIUS Tunnels

This chapter describes how to set up and administer RADIUS tunnels.

About RADIUS Tunnels

A tunnel is a uniquely secure type of remote connection. A tunnel passes data between a remote site and an enterprise site, providing an additional layer of encrypted protocol “wrapper” around the data. A tunnel offers authentication and encryption features that help secure the connection against network vandals and eavesdroppers. In addition, a tunnel can provide quality of service features such as guaranteed bandwidth.

NOTE: Steel-Belted Radius does not add tunnel functionality to your network. Steel-Belted Radius is able to support the authentication and accounting needs of any tunnels that you’ve already set up.

Administration and configuration of the tunnel happens at the remote site, since this is the side of the connection that requests remote access and opens the tunnel. An administrator at the remote site must configure the tunnel with various attributes: its destination IP address, what security protocols it supports, its password, and so on. These attributes are stored in a database to be retrieved when needed to set up a connection.

Storing tunnel attributes on a RADIUS server simplifies tunnel connections. At connection time, the tunnel is established by a network access device at the remote site. The NAD retrieves the tunnel configuration attributes from the RADIUS server and uses them to open the tunnel into the enterprise. After the tunnel is open, the user can be authenticated at the enterprise.

A RADIUS server is said to support tunnels if it has the ability to store and retrieve the configuration data that a NAD needs to open a tunnel. Steel-Belted Radius fully supports tunnels:

- Steel-Belted Radius can determine from the attributes in the incoming Access-Request whether the connection request involves a tunnel, and if so, which tunnel.
- Steel-Belted Radius can store and retrieve tunnel configuration data.
Steel-Belted Radius can track the number of tunnels currently in use, compare to a maximum number, and refuse the connection if the number is exceeded.

**Tunnel Authentication Sequence**

1. Steel-Belted Radius receives an Access-Request message:

2. Steel-Belted Radius checks if the Access-Request contains a Called-Station-Id attribute. If it does, Steel-Belted Radius searches its database for a tunnel entry that contains the indicated telephone number in its Called-Station-Id list.

   If a match between the Called-Station-Id and a tunnel entry can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

   **NOTE:** If realms are in use, Steel-Belted Radius also searches for this number in its realm configuration files. If a match is found, the Access-Request is routed to the realm, and the quest for a tunnel is abandoned. For this reason, it is important to ensure that DNIS numbers are unique across all tunnel entries and across all realm configuration files.

3. Steel-Belted Radius checks if the Access-Request contains a username in the form `User<Delimiter>TunnelName` or `TunnelName<Delimiter>User`. `<Delimiter>` is a single character that must match the server’s tunnel delimiter character. The order of the realm name relative to the username must match the server’s tunnel naming convention (prefix or suffix). Both of these values are determined per server (that is, all tunnels that use this server must follow the same conventions) by entering them in the Name Parsing tab of the Tunnels panel.

4. Steel-Belted Radius searches its database for a tunnel entry whose name matches the incoming `TunnelName`. If a match can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

5. If Steel-Belted Radius was able to match the Access-Request with a tunnel entry, the NAD uses the attributes returned in the Access-Accept message to open a tunnel into the enterprise site. Authentication of the User-Name is attempted, usually at the enterprise site. If user authentication succeeds, the connection is complete. Otherwise, the user’s connection request is denied.

6. If no matching tunnel entry was found in steps 1 or 2, Steel-Belted Radius concludes that a tunnel is not involved in making this connection. It then continues with its User-Name parsing sequence determine a destination for the authentication request.
**Configuring Tunnel Support**

To configure Steel-Belted Radius to support a tunnel, you must open the Tunnels panel (described on page 80) in the SBR Administrator and add a tunnel entry. A tunnel entry allows you to specify a list of connection Attributes such as the tunnel password, the IP address of the NAD at the enterprise site, encryption conventions to use, and so on. You can also enter the maximum number of tunnels that can be open at one time. You will need to coordinate with the administrator at the enterprise site to get some of this information.

**Called Station Id**

*DNIS (Dialed Number Information Services)* refers to a capability that many network access devices have to determine and use the telephone number that was dialed to make a connection request. The RADIUS standard supports DNIS by specifying the following attributes:

- **Calling-Station-Id** is the number from which the user originated the request.
- **Called-Station-Id** is the telephone number that was dialed to make the network connection.

When setting up a tunnel entry for the Steel-Belted Radius database, you can enter a telephone number or list of numbers in the **Called Station Id** list in the Tunnels panel. This list identifies Called-Station-Id attribute values that the server should expect to find in tunnel connection requests.

**Dictionaries for Tunnel Support**

The Tunnels panel allows you to create the Attributes list by selecting attributes from a drop-down list. The available selections include attributes from all standard and vendor-specific RADIUS dictionaries installed on the Steel-Belted Radius server. When the server can accept a tunnel connection request, it consults the corresponding tunnel entry for the list of Attributes to return in the Access-Accept packet. Steel-Belted Radius always returns any standard RADIUS attributes that appear in the Attributes list. It also returns any vendor-specific attributes that are appropriate for the NAD that requested the tunnel connection. Vendor-specific attributes in the Attributes list that do not apply to the requesting NAD are ignored.

**Concurrent Tunnel Connections**

Steel-Belted Radius tracks the number of active connections for each tunnel. You can limit the number of concurrent connections that can be open through a specific tunnel. When a user requests a new connection through a tunnel, Steel-Belted Radius compares the number of active connections in a tunnel to the maximum number of connections: if a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.
For concurrent connection limits to work, each NAD that can open a tunnel must be configured for RADIUS accounting and the same Steel-Belted Radius server must be specified for both authentication and accounting.

**NOTE:** Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Juniper Networks for more information.

### Configuring RADIUS Tunnels

The Tunnels panel (Figure 34) lets you configure Steel-Belted Radius to support tunnels. When you add a tunnel entry, you are not creating a tunnel; you are enabling Steel-Belted Radius to support an existing tunnel’s authentication and accounting needs and specifying how the server should parse tunnel names.

**Figure 34: Tunnels Panel: Tunnels Tab**

#### Adding a Tunnel

To add a Tunnel entry:

1. Choose **Tunnels > Tunnels** in the sidebar.
2. Click the **Add** button in the Steel-Belted Radius toolbar.

The Add Tunnel dialog (Figure 35) opens.
3. Enter the name of the tunnel name in the **Name** field.

Tunnel names do not need to match the actual node name of a client tunnel server. The name you assign to a tunnel must not match the name assigned to a proxy target, realm, or tunnel in your Steel-Belted Radius configuration.

4. Enter a description of the tunnel in the **Description** field.

Tunnel descriptions are used only for administrative purposes and do not affect tunnel connections. This field is typically used to identify the user or organization that uses the tunnel.

5. Associate attributes and values with the tunnel you are setting up.

When a tunnel is used to make a connection, the attributes associated with the tunnel are filtered according to the make/model of the RADIUS client used to establish the connection.

To associate attributes and values with a tunnel:

a. Click the **Add** button below the **Attributes** list.

   The Add Tunnel Attribute dialog (Figure 36) opens.

---

**Figure 35: Add Tunnel Dialog**

![Add Tunnel Dialog](image)
b. Select the attribute you want to add from the Attributes list.

c. Specify the string or IP address you want to use for the attribute value.

d. Click Add.

e. When you finish adding attributes for the tunnel, click Close.

6. Optionally, specify one or more Called Station IDs for the tunnel.

A Called Station ID is a telephone number that was dialed to make a network connection. The Called station ID list identifies the Called-Station-Id attribute values that the server expects to find in tunnel connection requests.

To add one or more Called Station ID numbers for a tunnel:

a. Click the New button to the right of the Called Station ID list.

The Add Called Station ID dialog (Figure 37) opens.

Figure 37: Add Called Station ID Dialog

b. Enter the number you want to use in the Called station ID field.

c. Click Add.

Repeat Steps a–c until you have added all called station IDs for the tunnel.

d. When you are finished adding called station IDs, click Close.

7. If you want to limit the number of connections that can use the tunnel simultaneously, click the Maximum open tunnels check box and enter the maximum number of tunnels in the Maximum open tunnels field.

8. Click OK.
**Editing a Tunnel**

To edit a tunnel entry:

1. Choose **Tunnels > Tunnels** in the sidebar and select the tunnel you want to edit. Click **Edit**.

The Edit Tunnel dialog (Figure 38) appears.

**Figure 38: Edit Tunnel Dialog**

2. Modify the settings for the tunnel as appropriate.

3. Refer to “Adding a Tunnel” on page 80 for information on how to use the fields and controls on the Edit Tunnel dialog.

4. When you are finished, click **Save**.

**Deleting a Tunnel**

To delete a tunnel entry from the Steel-Belted Radius database:

1. Choose **Tunnels > Tunnels** in the sidebar.

2. Select the tunnel you want to delete and click the **Delete** button on the Steel-Belted Radius toolbar (or right-click the entry and choose **Delete** from the context menu that appears).

3. When the Confirm Delete dialog opens, click **Yes**.
**Configuring Tunnel Name Parsing**

Tunnel name parsing lets Steel-Belted Radius determine whether the name string provided by a user includes a tunnel name by looking for the character configured as the delimiter for tunnel information. Tunnel name parsing options apply to all tunnels maintained by Steel-Belted Radius. You cannot set name parsing options for individual tunnels.

To configure tunnel name parsing:

1. Choose **Tunnels > Name Parsing** in the sidebar.

   The Name Parsing dialog (Figure 39) appears.

   **Figure 39: Tunnel Panel: Name Parsing Tab**

2. Click one of the following radio buttons:

   - **None**—Tunnel name parsing is not supported. If you choose this option, the tunnel authentication sequence is bypassed for each Access-Request; the server uses the standard username/password authentication sequence only.

   - **Tunnel name is prefix**—If the tunnel delimiter character is detected, the User-Name is assumed to be `TunnelName<PrefixDelimiter>User`.

   - **Tunnel name is suffix**—If the tunnel delimiter character is detected, the User-Name is assumed to be `User<SuffixDelimiter>TunnelName`.

   The option you choose applies to all tunnels defined on the server.

3. If you clicked **Tunnel name is prefix** or **Tunnel name is suffix**, use the Parsing delimiter field to specify the character used to separate the tunnel name and the username.

   The default delimiter character for tunnel name parsing is `@`.

   **NOTE:** Choose different delimiter characters and different prefix/suffix name parsing conventions for tunnels and for proxies or realms.
Chapter 9
Administering Address Pools

This chapter describes how to set up IPv4 and IPX address pools. Steel-Belted Radius does not support IPv6 address pools.

**NOTE:** Please contact Juniper Technical Support if you need address pools larger than 65,535 \(2^{16}\) addresses.

Address Pool Files

The following files establish settings for IP and IPX address pools. For more information about these files, refer to the *Steel-Belted Radius Reference Guide*.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>radius.ini</td>
<td>Specifies (among other things) the suffixes used to set up NAD-specific IP pools.</td>
</tr>
</tbody>
</table>

Setting Up IP Address Pools

The IP Address Pool and IPX Address Pool panels (Figure 40) allow you to set up one or more pools out of which unique IPv4 or IPX addresses are assigned as users require them. Each address pool consists of a list of one or more ranges of addresses.
Adding an IPv4 Address Pool

An IP address pool consists of one or more ranges of IPv4 addresses. You can add or delete ranges and set an optional description for each address pool.

To add an IP address pool:

1. Choose Address Pools > IP in the sidebar.
   The IP Address Pools panel appears.

2. Click the Add button in the toolbar.
   The Add IP Address Pool dialog (Figure 41) appears.

3. Enter the name of the IP address pool in the Name field.
4. Optionally, enter a description of the address pool in the **Description** field.

5. Identify the address range or ranges in the IP address pool.
   a. Click the **Add** button below the Address Ranges list.
      
      The Add IP Address Range dialog (Figure 42) opens.

      **Figure 42: Add IP Address Range Dialog**

      ![Add IP Address Range Dialog](image)

      b. Enter the first address in the **Starting address** field.

      c. Enter the number of addresses in the address range in the **Number of addresses** field.

      d. Click **Add**.

      e. Repeat steps a–d for each address range in the IP address pool. When you are finished, click **Close**.

6. Click **OK**.

**Editing an IP Address Pool**

To edit an IP address pool:

1. Choose **Address Pools > IP** in the sidebar.

2. Select the entry you want to modify and click the **Edit** button (or right-click the entry and choose **Edit**).

   The Edit IP Address Pool dialog (Figure 43) appears.

**Figure 43: Edit IP Address Pool Dialog**

![Edit IP Address Pool Dialog](image)
3. Modify the settings for the address pool as needed.

- To add an address range to the address pool, click the Add button and specify the starting address and number of addresses in the range.
- To modify an address range, select it and click the Edit button.
- To delete an address range from the address pool, select it and click the Delete button.

4. When you are finished, click Save.

**Removing an IP Address Pool**

To delete an IP address pool:

1. Choose Address Pools > IP in the sidebar.

2. Select the entry you want to remove and click the Delete button (or right-click the entry and choose Delete).

3. When you are prompted to confirm the deletion, click Yes.

**Specifying an IP Address Pool for User/Profile Records**

The Framed-IP-Address return list attribute controls how the server assigns an IP address to a user making a connection. When you add or edit the Framed-IP-Address attribute in the Users or Profiles dialog, the Add Attribute dialog (Figure 44) allows you to choose an IP address pool instead of specifying an IP address.

**Figure 44: Editing the Framed-IP-Address**

---

**Setting Up IPX Address Pools**

The IPX Pools dialog (Figure 45) allows you to set up one or more pools out of which unique IPX network numbers are assigned as users require them. Each pool consists of a list of one or more ranges of IPX network numbers.
Adding an IPX Pool

An IPX pool consists of one or more ranges of IPX network numbers. You can add or delete ranges and set an optional description for each address pool.

To add an IPX address pool:

1. Choose Address Pools > IPX in the sidebar.
   
   The IPX Address Pools panel appears.

2. Click the Add button in the toolbar.

   The Add IPX Address Pool dialog (Figure 46) appears.

3. Enter the name of the IPX address pool in the Name field.

4. Optionally, enter a description of the address pool in the Description field.

5. Identify the address ranges in the IP address pool.
   
   a. Click the Add button below the Address Ranges list.
The Add IP Address Range dialog (Figure 47) opens.

**Figure 47: Add IP Address Range Dialog**

![Add IP Address Range Dialog](image)

b. Enter the first address in the **Starting address** field.

c. Enter the number of IPX addresses in the address range in the **Number of addresses** field.

d. Click **Add**.

e. Repeat steps a–d for each address range in the IPX address pool. When you are finished, click **Close**.

6. Click **OK**.

**Editing an IPX Address Pool**

To edit an IPX address pool:

1. Choose **Address Pools > IPX** in the sidebar.

2. Select the entry you want to modify and click the **Edit** button (or right-click the entry and choose **Edit**).

The Edit IPX Address Pool dialog (Figure 48) appears.

**Figure 48: Edit IPX Address Pool Dialog**

![Edit IPX Address Pool Dialog](image)

3. Modify the settings for the address pool as needed.

- To add an address range to the address pool, click the **Add** button and specify the starting address and number of addresses in the range.
To modify an address range, select it and click the **Edit** button.

To delete an address range from the address pool, select it and click the **Delete** button.

4. When you are finished, click **Save**.

**Removing an IPX Address Pool**

To delete an IPX address pool:

1. Click the **Address Pools** button to display the Address Pools panel.

2. Click the **IPX Address Pools** tab to display the list of IPX address pools that have been configured.

3. Select the entry you want to remove and click the **Delete** button (or right-click the entry and choose **Delete**).

4. When you are prompted to confirm the deletion, click **Yes**.

**Specifying Pooled IPX Network Numbers in User/Profile Records**

The Framed-IPX-Address return list attribute controls how Steel-Belted Radius assigns an IPX address to a user making a connection.

When you add or edit the Framed-IPX-Address attribute for a user or profile, the Framed-IPX-Address dialog appears. To select an IPX address assignment option, type an IPX address in the **IPX address** field or click the **IPX Address Pool** check box and select the name of the IPX address pool you want to use from the list.

![Figure 49: Specifying an IPX Pool for the Framed-IPX-Address Attribute](image-url)
Chapter 10

Setting Up Administrator Accounts

This chapter describes how to set up Steel-Belted Radius administrators and specify what permissions an administrator holds.

Administrators Panel

The Administrators panel lets you grant and revoke the right to use the SBR Administrator to configure a Steel-Belted Radius server. Each time you log into a Steel-Belted Radius server, SBR Administrator prompts you to authenticate yourself by entering an account name and password.

Figure 50: Administrators Panel

When the Steel-Belted Radius software is installed, any user who is a member of the group Administrators on a Steel-Belted Radius server implicitly has the right to use the SBR Administrator at its default (full) level of access. The Administrators panel lets you modify these default permissions.

The Administrators panel lists the users and groups who have been explicitly granted the right to run the SBR Administrator. Local users or groups are shown with their normal name. Remote users or groups are shown with the name of the domain, followed by a backslash and then the name of the domain user or group.
Adding a Local Administrator

To add a local administrator to the Steel-Belted Radius database:

1. Choose **Administrators** in the sidebar.
   
The Administrators panel appears.

2. Click the **Add** button to open the Browse for Administrator dialog (Figure 51).

   **Figure 51: Browse for Administrator Dialogs**

3. Click the **Local** radio button to specify you are adding a local user.

4. Identify the local users or groups you want to add.
   - If you want to add a local user, click the **Users** tab and select the name of a user.
   - If you want to add a local group, click the **Groups** tab and select the name of a user group.

5. Click **Add**.

6. Continue adding local users and groups until you are done, then click **Close**.
Adding a Remote Administrator

NOTE: Browsing within a domain with a large number of users or groups to select an administrator or group name can adversely affect Steel-Belted Radius performance.

To grant access to a remote administrator within a domain:

1. Choose Administrators in the sidebar.
   The Administrators panel appears.

2. Click the Add button to open the Browse for Administrator dialog (Figure 51).

3. Click the Domain radio button to specify you are adding a remote (domain) user.

4. When the list of domains appears, select the domain within which you would like to grant access.

5. Identify the remote users or groups within that domain who you want to add.
   - If you want to add a remote user, click the Users tab and select the name of a user.
   - If you want to add a remote group, click the Groups tab and select the name of a user group.

6. Click Add.

7. Continue adding domain users and groups until you are done, then click Close.

Deleting an Administrator

To revoke rights for a Steel-Belted Radius administrator:

1. Choose Administrators in the sidebar.
   The Administrators panel appears.

2. Select the user or group whose administration rights you want to revoke.

3. Click the Delete button on the SBR Administrator toolbar (or right-click the entry and choose Delete).

4. When you are prompted to confirm the deletion, click Yes.

NOTE: Be careful not to revoke your own rights. If you do so, you will no longer have access to Steel-Belted Radius administrative functions.
Deleting an Administrator
Chapter 11
Setting Up Filters

This chapter describes how to set up and use filters in Steel-Belted Radius.

If you are running the Global Enterprise Edition (GEE) or Service Provider Edition (SPE) of Steel-Belted Radius, you can use a separately licensed add-on module to use Javascript to select and create filters. For more information, refer to the Steel-Belted Radius Scripting Guide.

Overview

A filter is a collection of rules for adding, modifying, or removing attributes or attribute values in RADIUS requests and responses. You define filters and their rules by means of the Add Filter dialog (described on page 102). You enable filters by referring to them by name when using the SBR Administrator or when editing certain .ini file sections.

A filter consists of one or more rules, which are processed in sequential order.

- **Add rules** specify that an attribute-value pair (AVP) is added to a RADIUS packet during processing. The AVP is added after all other rules are processed. An attribute is added to a packet only if it is legal to do so.

  Some attributes can appear only once in a RADIUS packet; others can appear multiple times. If an attribute that is the subject of an Add rule is already present in the packet (after processing Allow and Exclude rules) and the attribute can only appear once, the Add rule is not processed and the second instance of the attribute is not added.

- **Allow rules** specify whether an attribute (or AVP) is allowed in a RADIUS packet.

  - If an Allow rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are allowed in the RADIUS packet.

  - If an Allow rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are allowed in the RADIUS packet.

  - If an Allow rule does not specify an attribute name, then all attributes, regardless of value, are allowed in the RADIUS packet.
Exclude rules specify an attribute (or AVP) is excluded from a RADIUS packet.

- If an Exclude rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are excluded from the RADIUS packet.
- If an Exclude rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are excluded from the RADIUS packet.
- If an Exclude rule does not specify an attribute name, then all attributes, regardless of value, are excluded from the RADIUS packet.

Replace rules specify the conditions whereby one attribute (or attribute value) is replaced with another.

- If a Replace rule specifies that one named attribute of a specified value \((\text{attr1} v1)\) should be replaced with a different attribute of a specified value \((\text{attr2} v2)\), then any occurrence of the first AVP is replaced with the second AVP. Result: \(\text{attr2} v2\).
- If a Replace rule specifies that a named attribute without a specified value \((\text{attr1})\) should be replaced with a different attribute of a specified value \((\text{attr2} v2)\), then any occurrence of the first attribute (regardless of value) is replaced with the second AVP. Result: \(\text{attr2} v2\).
- If a Replace rule specifies that one named attribute of a specified value \((\text{attr1} v1)\) should be replaced with a different attribute without a specified value \((\text{attr2})\), then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: \(\text{attr2} v1\).
- If a Replace rule specifies that one named attribute (without a specified value) should be replaced with a different attribute without a specified value, then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: \(\text{attr2} v1\).

Script rules specify when to run attribute filter scripts. For information on attribute filter scripts, refer to the Steel-Belted Radius Scripting Guide.

The Steel-Belted Radius dictionary file `radius.dct` provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.

**NOTE:** Filter rules provide you with tremendous flexibility. However, Steel-Belted Radius does not prevent you from creating an invalid RADIUS packet. Some attributes are not appropriate for certain types of requests. For example, adding a pooled Framed-Ip-Address attribute to an accounting request could cause a loss of available IP addresses.
Order of Filter Rules

The order of rules within a filter is important. General default rules that take no parameters, such as Allow (allow all attributes unless otherwise specified) or Exclude (exclude all attributes unless otherwise specified) must appear as the first rule in the filter. Later rules supersede earlier rules; the last applicable rule “wins.” Add and Replace rules are applied after the Allow and Exclude rules.

More specific rules with more parameters (Add attribute value) act as exceptions to less specific rules with fewer parameters (Allow attribute, EXCLUDE). For example, you might want to allow a certain attribute and exclude one or more specific values for that attribute. Or you might exclude all attributes, allow specific attributes, and add specific attribute/value pairs.

You can use two basic approaches to designing a filter:

- Start the rule list with a default Exclude rule (no parameters) and add Allow rules for any attributes or attribute/value pairs that you want to insert into the packet. Add and Replace rules may be used.
- Start the rule list with a default Allow rule (no parameters) and add Exclude rules for any attributes or attribute/value pairs that you want to remove from the packet. Add and Replace rules may be used.

The default action for Steel-Belted Radius is Exclude. If a filter does not contain any rules, the filter removes all attributes from a packet when the filter is applied.

Values in Filter Rules

The value of an attribute is interpreted based on the type of the attribute in its attribute dictionary. Table 14 lists the meaning of each attribute type.

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>hexadecimal</td>
<td>A hexadecimal value is specified as a string. Special characters may be included using escape codes.</td>
</tr>
<tr>
<td>int1, int4, integer</td>
<td>1- or 4-byte unsigned decimal number (integer is equivalent to int4). NOTE: The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.</td>
</tr>
<tr>
<td>ipaddr, ipaddr-pool</td>
<td>An IP address in dotted notation; for example: EXCLUDE NAS-IP-Address 127.0.0.1</td>
</tr>
<tr>
<td>ipxaddr-pool</td>
<td>A sequence of hex digits; for example: ALLOW Framed-IPX-Network 0042A36B</td>
</tr>
</tbody>
</table>
Referencing Attribute Filters

Steel-Belted Radius attribute filtering provides flexibility in packet processing. You reference filters by name in SBR Administrator dialogs, in various .ini and .aut configuration files, and in the FilterOut and FilterIn sections of your .pro and .dir files. You can use the same filter for all packets in all realms. You can apply filtering to some realms, and not others.

To disable filtering for a realm, omit filtering parameters from the * .pro or *.dir files and from the EAP-PEAP/EAP-TTLS configurations. Filtering is often used only for packets that are routed “out” to realms (the FilterOut parameter).
Chapter 11: Setting Up Filters

To reference filtering rules in proxy or directed realm configurations, you must use the FilterOut and FilterIn parameters in the [Auth] and [Acct] sections of a realm configuration file. For more information, refer to the Steel-Belted Radius Reference Guide.

NOTE: Do not allocate IP addresses from Steel-Belted Radius IP address pools in accounting filters. These addresses will be allocated but never released.

Filters Panel

You can use the Filters panel (Figure 52) to display the filters configured for Steel-Belted Radius. To open the Filters panel, click Filters in the SBR Administrator sidebar.

Figure 52: Filters Panel

Adding a Filter

To add a filter:

1. Click Filters in the SBR Administrator sidebar to display the Filter panel.

2. Click Add in the toolbar to display the Add Filter dialog (Figure 53).
Adding a Filter

3. Enter the name of the filter in the **Name** field.

4. Optionally, enter a description of the filter in the **Description** field.

   You can enter as many as 4095 characters for a filter description.

5. Use the **Default Rule** radio buttons to specify whether attributes should be allowed or excluded if no other rule applies to a RADIUS packet.

6. Click the **Add** button to display the Add Rule dialog.

**Figure 54: Add Rule Dialog**

7. Select the type of rule you want to add to the filter.

   Options are **Add**, **Allow**, **Exclude**, **Replace**, and **Script**. The fields in the Add Rule dialog may change depending on the option you select.

8. Specify the attribute name and value settings you want to use for the rule.
Searching the Filter List

You can search your list of filters to identify those of a specific type or that use a specific rule. To search your filter list:

1. Open the Filters panel.
2. Click the Search toolbar button.
3. When the Search Filters dialog (Figure 55) appears, enter one or more search criteria in the fields provided.

You can use the question mark (?) wildcard to represent one character in a string. You can use the asterisk (*) wildcard to represent any number of characters. For example, entering the search string MS-MPPE-* would identify any filter that looks for the MS-MPPE-Recv-Key or MS-MPPE-Send-Key attribute.

Figure 55: Search Filters Dialog

4. Click OK.

The Search Results dialog (Figure 56) displays the list of filters that satisfy your search criteria.
Figure 56: Search Results Dialog
Chapter 12

Setting Up EAP Authentication Policies

This chapter presents an overview of concepts relating to the Extensible Authentication Protocol (EAP) and describes how to configure Steel-Belted Radius to use EAP authentication methods and plug-ins.

About the Extensible Authentication Protocol

Steel-Belted Radius supports the Extensible Authentication Protocol (EAP), a standard for communication between network access devices and servers that provides for the future extensibility of authentication protocols.

EAP allows specialized knowledge about authentication protocols to be taken out of a NAD so that it acts solely as a conduit between authentication server and client. This means that new types of authentication can be supported by adding the appropriate functionality to server and client, without any changes to PPP or network access devices. When the authentication process is complete, the RADIUS server simply informs the NAD of the result.

Steel-Belted Radius supports several EAP authentication mechanisms, such as TTLS, TLS, FAST, PEAP, LEAP, MD5-Challenge, and Generic Token. Support for EAP has been designed to anticipate other authentication types as they become available.

For technical details about EAP, see RFC 2284, “PPP Extensible Authentication Protocol (EAP),” and RFC 2869, “RADIUS Extensions.”

Handling EAP Requests

The flow of RADIUS packets in an EAP scenario is quite different from the transactions using standard user credentials (for example, PAP or CHAP). Standard user credentials involve the transmission of a RADIUS request from the NAD to Steel-Belted Radius and a response (either an Accept or Reject) from the server back to the NAD.

With EAP, the first packet sent from the NAD to Steel-Belted Radius contains an EAP-Message attribute containing an EAP Identity Response. This is a signal sent by the system being authenticated that it wants to be authenticated by means of EAP. It is now up to Steel-Belted Radius to select the EAP protocol with which it is to authenticate the end-user.
The contents of the User-Name attribute is the only guideline available to Steel-Belted Radius in selecting the appropriate EAP protocol. Should Steel-Belted Radius select an EAP protocol that is not supported by the client, the client has the opportunity to send an EAP-NAK and to request a specific alternate protocol.

**NOTE:** Given this general flow, a RADIUS request with EAP credentials must incur a minimum of two network round-trips between the RAS (or Access Point) and the Steel-Belted Radius before reaching a successful conclusion.

**Automatic EAP Helpers**

Automatic EAP helpers serve as intermediaries between EAP and traditional authentication methods. These helper modules may be configured (using an associated .eap file) to work with existing authentication methods to shield the authentication methods from the particulars of the selected EAP protocol.

Table 15 indicates whether each EAP type is implemented as an EAP helper or stand-alone module in Steel-Belted Radius.

**Table 15: EAP Implementations**

<table>
<thead>
<tr>
<th>EAP-Type</th>
<th>Implemented As</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAP-FAST</td>
<td>Standalone Authentication Method Module</td>
</tr>
<tr>
<td>EAP-TTLS</td>
<td>Standalone Authentication Method Module</td>
</tr>
<tr>
<td>EAP-TLS</td>
<td>Standalone Authentication Method Module</td>
</tr>
<tr>
<td>EAP-TLS</td>
<td>Automatic EAP helper</td>
</tr>
<tr>
<td>LEAP</td>
<td>Automatic EAP helper for MS-CHAP-v1</td>
</tr>
<tr>
<td>EAP Generic-Token</td>
<td>Standalone Authentication Method Module (SecurID)</td>
</tr>
<tr>
<td>EAP MD5-Challenge</td>
<td>Automatic EAP helper for CHAP</td>
</tr>
<tr>
<td>EAP MS-CHAP-v2</td>
<td>Automatic EAP helper for MS-CHAP-v2 (needed for PEAP)</td>
</tr>
</tbody>
</table>

Whether an automatic EAP helper can be used in conjunction with a specific authentication method depends on what types of credentials the authentication method supports.

The automatic EAP helper that implements EAP MD5-Challenge generates CHAP credentials, while the helper that implements LEAP generates MS-CHAP-v1 credentials. As such, EAP MD5-Challenge can be used only with authentication methods that support CHAP, and LEAP can be used only with authentication methods that support MS-CHAP-v1.
Table 16 summarizes the support for MS-CHAP-v1 and CHAP in the Steel-Belted Radius authentication methods.

**Table 16: MS-CHAP-v1 and CHAP Support**

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>MS-CHAP-V1</th>
<th>CHAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind</td>
<td>Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind</td>
</tr>
<tr>
<td>Local</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy RADIUS</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SecurID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL</td>
<td>Yes if password is in clear or encrypted using enc-md5 in SQL database</td>
<td>Yes if password is in clear or encrypted using enc-md5 in SQL database</td>
</tr>
<tr>
<td>TACACS +</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>UNIX User</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>UNIX Group</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Windows Domain User</td>
<td>Yes (server must be running under SYSTEM account)</td>
<td>No</td>
</tr>
<tr>
<td>Windows Domain Group</td>
<td>Yes (server must be running under SYSTEM account)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Authentication Request Routing**

The order in which authentication methods and automatic EAP helpers are called to handle an authentication request depends on two factors:

- The ordered list of enabled authentication methods (viewable in the Authentication Policies panel in SBR Administrator). Refer to “Activating EAP Methods” on page 149 for information on using the Authentication Policies panel.
- The EAP-related configuration for each of the enabled authentication methods in the eap.ini file, which you configure from the Authentication Policies panel.

When Steel-Belted Radius receives an authentication request that does not contain EAP credentials, it passes the request to each enabled authentication method until one of the methods claims the request.

The EAP settings in the eap.ini file come into play only when a request with EAP credentials is received. An authentication request contains EAP credentials if it includes one or more EAP-Message attributes and contains no other form of user credentials (for example, User-Password).
EAP-Only Setting

When an authentication method’s EAP-Only setting is 1, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials. Under this setting, the authentication method is also bypassed if an authentication request specifically requests an EAP protocol that is not listed in the authentication method’s EAP-Type list in the eap.ini file.

NOTE: The PEAP authentication method plug-in converts the inner EAP/Generic Token credentials to PAP for security reasons. If you are using SecurID with PEAP, you should set the EAP-Only setting to 0.

First-Handle-Via-Auto-EAP Setting

If your configuration involves clients using more than one EAP protocol, Steel-Belted Radius must select an initial EAP protocol with which to proceed when receiving an authentication request with EAP credentials.

Selecting the incorrect EAP protocol is not fatal; the client simply sends an EAP NAK in response to the server’s selected protocol and suggests an alternate one. After one additional network round-trip, the correct EAP protocol becomes active.

Depending on the capabilities of the authentication methods being used, you may be able to cut out this additional network round-trip that affects a portion of your EAP-based authentication requests.

If an authentication method can check for the existence of a user and can retrieve the user’s password information with only the information available in the authentication request (for example, the username), it is said to be prefetch-capable. A prefetch-capable authentication method could be consulted first to see if a user exists in its database before committing to a specific EAP protocol.

If your authentication method is prefetch-capable, you would set First-Handle-Via-Auto-EAP to 0, indicating that the authentication method should have the first chance to handle the request. You would also set First-Handle-Via-Auto-EAP to 0 if the authentication method is capable of handling EAP credentials all on its own (clearly, it would not expect an automatic helper EAP method to do work on its behalf in this case).

By configuring the authentication method to be called first, Steel-Belted Radius can delay selection of an EAP protocol until it has ascertained whether the user exists in a particular authentication method’s database. This is a useful technique when you plan to use more than one EAP protocol, but you do not know which one the client will want. Even in this scenario, automatic EAP helpers may still end up performing the EAP protocol processing; they will take over after the authentication method has retrieved a user’s password information, rather than before.

The goal of an automatic EAP helper is to generate credentials against which traditional authentication methods (ones that do not understand EAP) can operate. Once an automatic EAP helper has generated these credentials, the authentication method that triggered the use of the helper is checked first for a password/credential match. Should this match not be present, the same traditional credentials are passed to all remaining enabled authentication methods in the master list (in the order in which they appear in the list).
EAP-NAK Notifications

If you are supporting only one type of client or only one EAP protocol, Steel-Belted Radius selects that EAP protocol for all EAP-based authentication requests it receives. If you are planning to support multiple EAP protocols and do not intend to maintain databases that track the appropriate EAP protocol on a user-by-user basis, Steel-Belted Radius automatically selects the appropriate EAP protocol for you.

When multiple EAP protocols are in play, you should configure each authentication method you plan to use with all the EAP protocols that may be used with it. In this configuration, when Steel-Belted Radius receives an authentication request containing EAP information, it chooses the first EAP protocol listed for the first authentication method that claims the request. Should the client require a different EAP protocol, it sends back an EAP-NAK that specifies the EAP protocol it would prefer to use.

After receiving an EAP-NAK, Steel-Belted Radius performs a scan of the authentication methods, in search of the first authentication method that has the requested EAP protocol listed (the authentication method may support this EAP protocol directly or with the help of an automatic EAP helper).

If the requested EAP protocol does not appear in any of the authentication methods’ lists of supported EAP protocols, Steel-Belted Radius rejects the authentication request.

### Table 17: Authentication Method Prefetch Capability

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Prefetch Capable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>Yes, if using BindName (rather than the Bind option)</td>
</tr>
<tr>
<td>Native User</td>
<td>Yes</td>
</tr>
<tr>
<td>SQL</td>
<td>Yes, if password does not need to be used as an input parameter in the SQL statement</td>
</tr>
<tr>
<td>UNIX User</td>
<td>No</td>
</tr>
<tr>
<td>Windows Domain</td>
<td>No</td>
</tr>
</tbody>
</table>

**NOTE:** If you enable the lockout facility in Steel-Belted Radius and you use a tunneled authentication method (TTLS or PEAP) with a prefetch-capable method (native user, SQL, or LDAP) and an enabled EAP protocol (MS-CHAPv2, MD5-Challenge, LEAP, TLS), then you must enable First Handle via Auto-EAP in that prefetch-capable method to prevent the outer username (anonymous) from being added to the lockout list.

Otherwise, when Steel-Belted Radius receives an authentication request that uses an unconfigured EAP method, Steel-Belted Radius will reject the user (because the EAP method is not configured) and add the outer username (anonymous) to its lockout list. This will result in all users with an outer authentication name of anonymous being rejected until the lockout period expires.

**EAP-NAK Notifications**

If you are supporting only one type of client or only one EAP protocol, Steel-Belted Radius selects that EAP protocol for all EAP-based authentication requests it receives. If you are planning to support multiple EAP protocols and do not intend to maintain databases that track the appropriate EAP protocol on a user-by-user basis, Steel-Belted Radius automatically selects the appropriate EAP protocol for you.

When multiple EAP protocols are in play, you should configure each authentication method you plan to use with all the EAP protocols that may be used with it. In this configuration, when Steel-Belted Radius receives an authentication request containing EAP information, it chooses the first EAP protocol listed for the first authentication method that claims the request. Should the client require a different EAP protocol, it sends back an EAP-NAK that specifies the EAP protocol it would prefer to use.

After receiving an EAP-NAK, Steel-Belted Radius performs a scan of the authentication methods, in search of the first authentication method that has the requested EAP protocol listed (the authentication method may support this EAP protocol directly or with the help of an automatic EAP helper).

If the requested EAP protocol does not appear in any of the authentication methods’ lists of supported EAP protocols, Steel-Belted Radius rejects the authentication request.
Reauthenticating Connections

Most Access Points understand only a limited number of attributes that may be included in a RADIUS response to signal that the user has been accepted. The **Session-Timeout** attribute is of particular significance in a WLAN realm as it instructs the Access Point how long to allow the user to remain connected to a WLAN before having to re-authenticate to Steel-Belted Radius.

You can configure your choice of **Session-Timeout** settings using standard Steel-Belted Radius reply-list items on a user-by-user basis. If you are using EAP-TLS or EAP-TTLS to authenticate users, you can also have these modules automatically generate **Session-Timeout** attributes based on policies set in their configuration files. This level of control is necessary for EAP-TLS and EAP-TTLS as these modules also support **session resumption**, a quicker method of re-authenticating users. The value in the **Session-Timeout** attribute may need to be dynamically calculated in these cases.

**NOTE:** Not all Access Points support the Session-Timeout attribute. You should check your Access Points’ specifications to determine whether this configuration must be performed in a fixed manner on the Access Point or if the Access Point should defer to the server.

Certificates

A certificate is an electronic data structure used to identify an individual, a server, a company, or some other entity, and to associate that identity with a public key and an associated private key. Like a passport, a certificate provides generally recognized proof of an entity’s identity. Certificates bind public key values to entities, so that remote users of an entity’s public key can be certain the associated private key is owned by the correct person or system. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate. The most widely accepted format for certificates is defined by the ITU-T X.509 international standard, which is described in RFC 3280, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.”

**Certificate authorities (CAs)** are entities that validate identities and issue certificates. An organization that wants to serve as its own CA can issue its own certificates, or an organization can purchase certificates from a trusted third-party CA. The methods used to validate an identity vary depending on the policies of a given CA. In general, before issuing a certificate, a CA must verify the identity of the entity and must digitally sign the certificate to ensure it cannot be modified. This ensures that a certificate issued by a CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee).

In addition to a public key, a certificate includes the name of the entity it identifies, an expiration date, the name of the CA that issued the certificate, a serial number, and the digital signature of the issuing CA, which creates a mathematical relationship between the signing CA certificate’s public key and the public key of the certificate it signs. The CA’s digital signature allows the certificate to function as a “letter of introduction” for users who know and trust the CA but don’t know the entity identified by the certificate.
Because a certificate’s expiration date is part of its signed contents, remote entities can verify that a certificate is valid and current.

Common types of certificates include the following:

- **Certificate Authority certificates** can sign other certificates.

- **Server certificates** are used on a server to enable a software client to verify the validity of the connection to a machine ("Am I really connecting to www.juniper.net?") and to create an encrypted channel between a client and a server.

- **Client certificates** are used to allow a server to verify a client’s identity (certificate based authentication) and to allow a user to digitally sign or encrypt data.

### Certificate Chains

A certificate chain is a sequence of certificates, where each certificate in the chain is signed by the certificate above it in the chain. At the top of the chain is a self-signed certificate. Each CA in the chain vouches for the identity in the entity to which it issues a signed digital certificate. Certificate chains establish a chain of trust; if you trust the CA at the top of the chain, this implies you can trust the signed certificates below it in the chain.

### Certificate Revocation Lists

Under normal circumstances, a certificate remains valid until it reaches its expiration date. However, a certificate may become invalid before it expires. For example, if an employee whose identity is bound to a certificate terminates employment or if an enterprise suspects the confidentiality of the private key associated with a certificate’s public key has been compromised, the certificate may be declared invalid and revoked.

When a CA revokes a certificate, it must let other entities know the certificate is no longer valid and should not be accepted. A **Certificate Revocation List** is a signed data structure that identifies the serial numbers of certificates that have been issued and subsequently revoked by the CA. When a remote entity is asked to use a certificate to verify a remote user’s identity, it can download a current copy of the applicable CRL and confirm that the certificate’s serial number is not present.

CRLs can be issued by a CA on periodically (hourly, daily, or weekly) or as needed. When a certificate is revoked, its serial number is listed in the CRL, and that serial number remains in the CRL at least one period after the certificate’s expiration date. CRLs, like certificates, can be distributed by untrusted servers and untrusted communications.

Under some circumstances, **latency** (the time between when a certificate is revoked and when the certificate’s serial number appears on the CRL of the issuing CA) may be a concern. For example, if a revocation is reported today, that revocation will not be reliably notified to certificate-using systems until all currently issued CRLs are updated, which may take hours, days, or even weeks. Online revocation checking can reduce the latency between a revocation report and the distribution of the information to relying parties.
If CRL checking is enabled, Steel-Belted Radius uses the information contained in a client certificate to connect to the certificate’s CRL Distribution Point (CDP). Steel-Belted Radius then uses HTTP, LDAP, or a network file system to retrieve the appropriate CRLs. Steel-Belted Radius stores these retrieved CRLs in the CRLCache directory under the radiusdir server directory.

When a client certificate is presented during EAP-TLS authentication, Steel-Belted Radius evaluates the client’s certificate chain against its set of stored CRLs to verify none of the certificates in the chain have been revoked.

You can configure the following settings for CRL checking:

- CRL expiration—The CRL checking feature can be configured to operate in strict or lax mode.
  - In strict mode, a cached CRL that has expired will be immediately discarded; if Steel-Belted Radius cannot acquire a new CRL in the allotted time during a CRL check on a chain, the user is rejected.
  - In lax mode, you can configure Steel-Belted Radius to accept an expired CRL for a a period past its expiration.

Note that Steel-Belted Radius attempts to obtain a current CRL whether it is running in strict or lax mode.

- Missing CDP attribute—When a CRL check is performed on a certificate chain, Steel-Belted Radius reads the contents of the CDP attribute for each certificate past the root certificate and uses the CDP information to retrieve the appropriate CRL. If a non-root certificate in the chain does not contain a CDP attribute, no CRL checking will be performed for that certificate. You can configure EAP-TLS to reject the user if it encounters a non-root certificate that is missing a CDP attribute.

- Incomplete LDAP CDP—Some CAs may create certificates that contain an LDAP-style CDP (//ldap:\... that does not specify the identity of the LDAP server to be queried. You can designate a default LDAP server that will be used when such CDPs are encountered. If you do not designate a default LDAP server and an LDAP-style CDP is encountered, the CRL retrieval will fail.

---

**EAP-TLS**

The EAP-TLS (Transport Layer Security) protocol requires that both user and authentication server have certificates for mutual authentication. While the mechanism is very strong, it requires that the corporation that deploys it maintain a certificate infrastructure for all of its users.

EAP-TLS can be deployed as an authentication method or as an automatic EAP helper.

- When EAP-TLS is deployed as an authentication method, EAP-TLS appears in the Authentication Policies panel in SBR Administrator after it is deployed as an authentication method. You can use the Authentication Policies panel to enable the EAP-TLS method and specify its sequence relative to other authentication methods Steel-Belted Radius uses.
When EAP-TLS is deployed as an authentication method, you can configure it to perform certificate revocation list (CRL) checking. When CRL checking is enabled, EAP-TLS confirms that the client’s certificate chain traces back to one of the trusted root certificates installed at initialization and checks the serial number of each certificate in the chain against the contents of CRLs to verify that none of the certificates in the chain have been revoked.

You can configure the `tlsauth.aut` file to call a fixed profile when TLS-EAP is used. This profile specifies the attributes that are sent back in response to a successful authentication.

You cannot use secondary authorization when EAP-TLS is deployed as an authentication method.

When EAP-TLS is deployed as an automatic EAP helper, you must list TLS in the EAP-Type list of an authentication method. When EAP-TLS is triggered, the `tlsauth` authentication goes through the TLS handshake required by the EAP-TLS specification. Assuming the user provides a certificate that the server can verify against a list of trusted root certificates, the EAP-TLS part of the exchange concludes successfully.

You may not want to grant access to your network to every user with a trusted certificate. By enabling the optional secondary authorization feature of the `tlsauth` plugin, you can have Steel-Belted Radius authorize users with valid certificates on a case-by-case basis. Secondary authorization also allows you to include user-specific attributes in an Access-Accept response; these attributes can be used to communicate options that are to be active for a user’s connection to the NAD. Without secondary authorization, the only attributes returned on an Access-Accept are those generated by the `tlsauth` plug-in itself (`termination-action` and `session-limit`).

If you enable the TLS authentication method, secondary authorizations must be performed by local authentication methods (they cannot be proxied). The authentication method you select for secondary authorizations must be able to authenticate users in a single pass; it cannot challenge the authorization request and request additional information. The username employed during secondary authorization is derived from a field in the user’s certificate. Since a user’s certificate does not include a password, you must configure `tlsauth` to make the secondary authorization request with no password or with a fixed password.

If you configure secondary authorization with no password, your selected authentication method must be capable of handling requests that do not include passwords; the only authentication methods that support this style of authentication and ship with Steel-Belted Radius are Native User, LDAP and SQL. If you configure secondary authorization with a fixed password, you can use any authentication method that supports PAP authentication. In this configuration all user records must have the same fixed password.
You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-TLS authentication method. For information on configuring your server certificate, see “Configuring Server Certificates” on page 145.

To configure EAP-TLS as an authentication method:

1. Select **Authentication Policies > EAP Methods** to open the EAP Methods panel (Figure 57).

   **Figure 57: EAP Methods Panel**

   ![EAP Methods Panel](image)

2. Click the **Enable** check box for the EAP-TLS authentication method.

3. Select the **EAP-TLS** entry and click the **Edit** button on the toolbar (or double-click the **EAP-TLS** entry).

   The Edit TLS Authentication Method dialog (Figure 67) opens.

   **Figure 58: Edit TLS Authentication Method dialog**

   ![Edit TLS Authentication Method dialog](image)
4. Use the tabs in the Edit TLS Authentication Method dialog to configure the following settings:

- Client certificate validation
- Session resumption
- Advanced server settings

Each configuration task is described separately below.

**Configuring Client Certificate Validation**

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS Authentication Method dialog.

2. Click the **Enable CRL Checking** check box to enable CRL checking.

3. Enter the number of seconds that EAP-TLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user’s authentication request results in a reject.

4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   EAP-TLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), EAP-TLS does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), EAP-TLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.

5. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

   Without a CDP attribute, EAP-TLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   If you click the **Allow Missing CDP Attribute** check box, EAP-TLS allows such certificates and skips CRL checking for them.

   If you clear the **Allow Missing CDP Attribute** check box, EAP-TLS does not accept a CRL with a missing CDP attribute.
6. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string \\ldap:\\ in the **Default LDAP Server Name** field.

CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

If you don’t specify a server name and such certificates are encountered, the CRL retrieval fails.

### Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

**NOTE:** For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Session Resumption** tab in the Edit TLS Authentication Method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session Timeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationally cheap.

3. Enter the integer value that you want returned in a Termination-Action attribute in the **Termination Action** field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached.

   If you enter 0 or if you do not specify a value for this attribute, the EAP-TLS method does not generate such an attribute. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.
4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

**Configuring Advanced Server Settings**

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TLS protocol:

1. Click the **Advanced Server Settings** tab in the Edit TLS Authentication Method dialog.

2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange, in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enable the **Return MPPE Keys** check box to specify whether the TLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

4. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

5. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.
These cipher suites are documented in RFC 2246, “The TLS Protocol Version 1,” and other TLS-related RFCs and draft RFCs.

Default value is 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09.

6. Optionally, enable the **Verify User Name is Principal** check box if you want Steel-Belted Radius to verify that the contents of the RADIUS User-Name attribute match the Principal Name of the certificate used to authenticate the user.

Certificates issued by Microsoft’s Windows 2000 Certificate Server typically include a Subject Alternative Name/Other Name attribute, where Principal Name set to something like `user@certtest.acme.com`.

The Windows XP client that supports EAP-TLS in conjunction with 802.1X extracts this attribute value from the client’s certificate and uses it to respond to the Access Point’s EAP Identity Request. The Access Point, in turn, packages up this value as the RADIUS User-Name attribute in requests it sends to a RADIUS server.

You should clear (disable) the check box if the certificates used do not include a Principal Name or if the client being used does not report the contents of Principal Name as the user’s identity in response to an EAP Identity Request.

### Configuring EAP-TLS as an Automatic EAP Helper

You must configure the server certificate for the Steel-Belted Radius server before you use the TLS EAP helper. For information on configuring your server certificate, see “Configuring Server Certificates” on page 145.

To configure EAP-TLS as an EAP helper:

1. Select **Authentication Policies > EAP Methods** to open the EAP Methods panel (Figure 59).

Figure 59: EAP Methods Panel
2. Click the **Enable** check box for the **EAP-TLS Helper** method.

3. Select the **EAP-TLS Helper** entry and click the **Edit** button on the toolbar (or double-click the **EAP-TLS Helper** entry).

The Edit TLS EAP Helper Method dialog (Figure 67) opens.

**Figure 60: Edit TLS EAP Helper Method dialog**

4. Use the tabs in the Edit TLS EAP Helper Method dialog to configure the following settings:
   - Client certificate validation
   - Secondary authorization
   - Session resumption
   - Advanced server settings

   Each configuration task is described separately below.

**Configuring Client Certificate Validation**

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS EAP Helper Method dialog.

2. Click the **Enable CRL Checking** check box to enable CRL checking.

3. Enter the number of seconds that the TLS EAP helper will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user's authentication request results in a reject.
4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

The TLS EAP helper always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

- If you enter 0 (strict expiration mode), the TLS EAP helper does not accept a CRL that has expired.
- If you enter a value greater than 0 (lax expiration mode), the TLS EAP helper considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.

5. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

Without a CDP attribute, the TLS EAP helper will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

If you click the **Allow Missing CDP Attribute** check box, the TLS EAP helper allows such certificates and skips CRL checking for them.

If you clear the **Allow Missing CDP Attribute** check box, the TLS EAP helper does not accept a CRL with a missing CDP attribute.

6. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string `//ldap:\\` in the **Default LDAP Server Name** field.

CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

If you don't specify a server name and such certificates are encountered, the CRL retrieval fails.

**Configuring Secondary Authentication**

Secondary authorization settings let you specify whether secondary authorization is performed and, if it is, what information is used in the secondary authorization request.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Secondary Authorization** tab in the Edit TLS EAP Helper Method dialog.

2. Click the **Enable Secondary Authorization** check box to enable secondary authorization checking.

   If secondary authorization is disabled, the EAP-TLS plug-in accepts the user upon proof of ownership of a private key that matches a valid certificate.

   If secondary authorization is enabled, a secondary authorization check against a traditional authentication method such as an SQL plug-in is performed.
3. Specify whether you want user names to be converted to Subject CN names or principal names.

After the EAP-TLS module has concluded its processing, it may still defer to a traditional authentication method (core or plug-in) for final authorization. To do so, it must provide a user name and password to the traditional authentication method.

- If you click the **Subject CN** option button, the EAP-TLS module parses the Subject attribute of the client's certificate for the least significant 'CN=' and takes the value of this attribute (for example, 'George Washington') as the user name being passed to the traditional authentication method.

- If you click the **Principal Name** option button, the EAP-TLS module uses the principal name (Subject Alternate Name or Other Name) from the client certificate (for example, **joe@acme.com**) as the user name being passed to the traditional authentication method.

4. If you plan to use secondary authorization against an authentication method (for example, LDAP) that cannot be configured to ignore the lack of user credentials, specify a fixed password that the plug-in uses on all secondary authorization checks in the **Fixed Password** field.

By default, the secondary authorization check includes a user name but no other user credentials, because no password or similar credential for the client is available at the conclusion of the TLS handshake. Some authentication methods (Native User, LDAP, and SQL) can be configured to not require user credentials.

5. If you want the EAP-TLS plug-in to add four attributes to the request before the secondary authorization check is performed, click the **Include Certificate Info** check box.

When the Include Certificate Info check box is clicked, Steel-Belted Radius adds the following attributes to the request:

- The Funk-Peer-Cert-Subject attribute contains the value of the Subject attribute in the client certificate.

- The Funk-Peer-Cert-Principal attribute contains the value of the principal name (Subject Alternate Name or Other Name) attribute of the client certificate.

- The Funk-Peer-Cert-Issuer attribute contains the value of the Issuer attribute in the client certificate.

- The Funk-Peer-Cert-Hash attribute contains a hexadecimal ASCII representation of the SHA1 hash of the client certificate.

These attributes are ignored if the authentication method that performs the authentication check does not use them.
Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

To configure session resumption for the TLS EAP helper protocol:

1. Click the Session Resumption tab in the Edit TLS EAP Helper Method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the Session Timeout field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationally cheap.

3. Enter the integer value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached.

   If you enter 0 or if you do not specify a value for this attribute, the TLS EAP helper method does not generate such an attribute. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the Resumption Limit field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.
Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the TLS EAP helper protocol:

1. Click the Advanced Server Settings tab in the Edit TLS EAP Helper Method dialog.

2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange, in the TLS Message Fragment Length field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the Max Transaction Time field.

   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.

4. Enable the Return MPPE Keys check box to specify whether the TLS EAP helper includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the DH Prime Bits list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the Cipher Suites field.

   These cipher suites are documented in RFC 2246, “The TLS Protocol Version 1,” and other TLS-related RFCs and draft RFCs.

   Default value is 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09.
EAP-TTLS

EAP-TTLS (Tunneled Transport Layer Security) is designed to provide authentication that is as strong as EAP-TLS, but it does not require that each user be issued a certificate. Instead, only the authentication servers are issued certificates. User authentication is performed by password; but the password credentials are transported in a securely encrypted “tunnel” established based upon the server certificates.

1. After the authentication server determines that the user has made an authentication request, it sends its certificate to the user’s system.

   **Figure 61: Server Certificate Sent to RAS**

   ![Diagram of initial contact between end user and server](image)

   - **Step 1: Initial Contact**
     - End User
     - NAS / Access Point
     - Server
     - Cert

2. The authentication server’s certificate is used to establish a tunnel between the user and the server.

   **Figure 62: Tunnel Established**

   ![Diagram of tunnel establishment between end user and server](image)

   - **Step 2: Establish Tunnel**
     - End User
     - NAS / Access Point
     - Server

3. Once the tunnel is established, credentials can be exchanged safely between the server and the user since tunnels encrypt all data in a secure fashion. This stage is called **inner authentication**.

   **Figure 63: Inner Authentication**

   ![Diagram of inner authentication between end user and server](image)

   - **Step 3: Inner Authentication**
     - End User
     - NAS / Access Point
     - Server
     - Credentials

With EAP-TTLS, it is not necessary to create a new infrastructure of user certificates. User authentication is performed against the same security database that is already in use on the corporate LAN; for example, Windows Domain Controllers, SQL or LDAP databases, or token systems.
Chapter 12: Setting Up EAP Authentication Policies

The routing of the inner authentication request can be handled via standard Steel-Belted Radius authentication request routing. If your EAP-TTLS tunnel ends at a dedicated server and all the inner authentication requests are to be performed by other servers, you should use standard request routing so the proxy realm target can be determined in a standard fashion (that is, the decoration of the username revealed by inner authentication).

**Configuring EAP-TTLS**

To configure EAP-TTLS as an authentication method:

1. Select **Authentication Policies > EAP Methods** to open the EAP Methods panel (Figure 64).

   **Figure 64: EAP Methods Panel**

   ![EAP Methods Panel](image)

2. Click the **Enable** check box for the **EAP-TTLS** method.

3. Select the **EAP-TTLS** entry and click the **Edit** button on the toolbar (or double-click the **EAP-TTLS** entry).

   The Edit TTLS Authentication Method dialog (Figure 67) opens.

   **Figure 65: Edit TTLS Authentication Method dialog**

   ![Edit TTLS Authentication Method dialog](image)
4. Use the tabs in the Edit TTLS Authentication Method dialog to configure the following settings:

- Request filters
- Response filters
- Client certificate validation
- Session resumption
- Inner authentication
- Advanced server settings

Each configuration task is described separately below.

**Configuring Request Filters**

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.

You must configure filters using the Filters panel before you can associate them with the EAP-FAST authentication method. For information on configuring filters, refer to Chapter 11, “Setting Up Filters” on page 97.

To configure request filtering for the EAP-TTLS protocol:

1. Click the Request Filters tab in the Edit TTLS Authentication method dialog.

2. Optionally, click the Transfer Outer Attribs to New check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests).

   - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   - If this filter is not specified, no attributes from the outer request are transferred to the inner request.

3. Optionally, click the Transfer Outer Attribs to Continue check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
Chapter 12: Setting Up EAP Authentication Policies

If this filter is not specified, no attributes from the outer request are transferred to the inner request.

4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

### Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-TTLS protocol:

1. Click the **Response Filters** tab in the Edit TTLS Authentication method dialog.

2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.

   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

3. Optionally, click the **Transfer Inner Attribs to Reject** check box and select the filter you want to use from the dropdown list.
This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

If this filter is not specified, no attributes from the outer request are transferred to the inner request.

**Configuring Client Certificate Validation**

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TTLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TTLS Authentication Method dialog.
2. Click the **Enable CRL Checking** check box to enable CRL checking.
3. If you want to require that the client must provide a certificate as part of the TTLS exchange, click the **Require Client Certificate** check box.
4. Enter the number of seconds that EAP-TTLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

When CRL retrieval takes longer than the specified time, the user’s authentication request results in a reject.

5. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

EAP-TTLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

- If you enter 0 (strict expiration mode), EAP-TTLS does not accept a CRL that has expired.
- If you enter a value greater than 0 (lax expiration mode), EAP-TTLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.

6. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

Without a CDP attribute, EAP-TTLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

- If you click the **Allow Missing CDP Attribute** check box, EAP-TTLS allows such certificates and skips CRL checking for them.
- If you clear the **Allow Missing CDP Attribute** check box, EAP-TTLS does not accept a CRL with a missing CDP attribute.
7. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\ in the Default LDAP Server Name field.

CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

If you don’t specify a server name and such certificates are encountered, the CRL retrieval fails.

Configuring Session Resumption
Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

NOTE: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TTLS protocol:

1. Click the Session Resumption tab in the Edit TTLS Authentication Method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the Session Timeout field.

   - If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   - If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationally cheap.

3. Enter the integer value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached.

   If you enter 0 or if you do not specify a value for this attribute, the EAP-TLS method does not generate such an attribute. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.
4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

**Configuring Inner Authentication Settings**

The inner authentication settings let you specify the way in which the inner authentication step is to operate.

To configure inner authentication settings for the EAP-TTLS protocol:

1. Click the **Inner Authentication** tab in the Edit TTLS Authentication Method dialog.

2. If you want requests to be routed based on the methods listed in the directed realm, enter the name of a directed realm in the **Directed Realm** field.

   Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. If you want requests to be processed by means of a realm selection script, enter the name of a script in the **Realm Selection Script** field.

**Configuring Advanced Server Settings**

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TTLS protocol:

1. Click the **Advanced Server Settings** tab in the Edit TTLS Authentication Method dialog.

2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange, in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.
If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.

4. Enable the **Return MPPE Keys** check box to specify whether the TTLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

These cipher suites are documented in RFC 2246, “The TLS Protocol Version 1,” and other TLS-related RFCs and draft RFCs.

Default value is `0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09`.

---

**EAP-FAST**

The EAP-FAST (Flexible Authentication via Secure Tunneling) protocol enables secure communication between a user and a server by using a TLS handshake to establish a mutually authenticated tunnel. Unlike EAP-TTLS and EAP-PEAP, EAP-FAST does not use a server certificate to secure the exchange. Instead, EAP-FAST uses a high-entropy secret called a Protected Access Credential (PAC) that is known to both the client and the AAA server to secure the TLS handshake.

Steel-Belted Radius generates PACs in a way that allows it to verify a PAC’s validity without having to store the individual PACs. This is accomplished by encrypting PACs with a server secret. By default, the EAP-FAST module creates and stores a new server secret every 30 days, though it retains retired server secrets until the PACs encrypted with those secrets expire.

EAP-FAST works in three phases:

- In Phase 0, a client without a valid PAC opens a connection through a RAS to the Steel-Belted Radius server. Steel-Belted Radius uses EAP-MS-CHAPv2 to authenticate the user. If authentication is successful, Steel-Belted Radius generates a PAC for the user and returns an Access-Reject message that contains the PAC.

If a client already has a valid PAC, Phase 0 is omitted.
In Phase 1, the client and the Steel-Belted Radius server establish a TLS tunnel based on the PAC presented by the user. During Phase 1, the server verifies that the PAC presented by the client was generated by its current or retired server secret.

In Phase 2, the server authenticates the user or machine credentials using EAP-GTC inside the protected TLS tunnel. If the PAC is valid and the user’s credentials are correct, the authentication succeeds and Steel-Belted Radius returns an Access-Accept message. If the user’s PAC is due to expire soon, Steel-Belted Radius provisions the client with a new PAC over the secure network connection at the end of Phase 2.

**NOTE:** Steel-Belted Radius converts the EAP-GTC inner authentication to Password Authentication Protocol (PAP) format, and it converts the inner EAP-MS-CHAPv2 authentication to MS-CHAPv2.

If the user’s password has expired and if the user’s password is stored in a backend server that supports password updates, such as Active Directory, then the PAC provisioning exchange can be used to change the user’s password. The password change concludes in the same exchange as the PAP provisioning.

The default lifetime for a user’s PAC is 90 days. By default, if Steel-Belted Radius encounters a PAC that will expire within the next 30 days, it provisions a new PAC as part of Phase 2 of the user authentication process. Provisioning a user with a new PAC while the user has a valid (unexpired) PAC is more efficient than issuing a new PAC, since it does not require a failed authentication and Access-Reject message to deliver a provisioned PAC.

If a user presents an expired PAC to Steel-Belted Radius, it ignores the expired PAC and initiates the PAC provisioning sequence starting at Phase 0.

**Configuring EAP-FAST**

You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-FAST authentication method. For information on configuring your server certificate, see “Configuring Server Certificates” on page 145.

To configure EAP-FAST on a Steel-Belted Radius server:

1. Select **Authentication Policies > EAP Methods** to open the EAP Methods panel (Figure 66).
2. Click the Enable check box for the EAP-FAST authentication method.

3. Select the EAP-FAST entry and click the Edit button on the toolbar (or double-click the EAP-FAST entry).

The Edit FAST Authentication Method dialog (Figure 67) opens.

4. Use the tabs in the Edit FAST Authentication Method dialog to configure the following settings:

- Request filters
- Response filters
- FAST protocol
- Inner authentication
- Advanced FAST server settings
Each configuration task is described separately below.

**Configuring Request Filters**

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.

---

You must configure filters using the Filters panel before you can associate them with the EAP-FAST authentication method. For information on configuring filters, refer to Chapter 11, “Setting Up Filters” on page 97.

---

To configure request filtering for the EAP-FAST protocol:

1. Click the **Request Filters** tab in the Edit FAST Authentication method dialog.

2. Optionally, click the **Transfer Outer Attribs to New** check box and select the filter you want to use from the dropdown list.
   
   This filter affects only a new inner authentication request (rather than continuations of previous requests).
   
   - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
   
   - If this filter is not specified, no attributes from the outer request are transferred to the inner request.

3. Optionally, click the **Transfer Outer Attribs to Continue** check box and select the filter you want to use from the dropdown list.
   
   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
   
   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.
   
   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.
   
   If this filter is not specified, the request remains unaltered.
5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

If this filter is not specified, the request remains unaltered.

### Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-FAST protocol:

1. Click the **Response Filters** tab in the Edit FAST Authentication method dialog.

2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.

   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

3. Optionally, click the **Transfer Inner Attribs to Reject** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

### Configuring FAST Protocol Settings

To configure protocol settings for the EAP-FAST protocol:

1. Click the **FAST Protocol** tab in the Edit FAST Authentication method dialog.
2. Enter the number of days that a server secret remains valid in the **Server Secret Lifetime** field.

   A new server secret is created and stored by the EAP-FAST plug-in when the current secret expires. (Old secrets are retained until all PACs encrypted with the old secret have expired.) Server secrets are stored in the `eapfast.info` file.

3. Enter the number of days that PACs will be accepted in the **PAC Lifetime** field.

4. Enter the number of days before the expiration of a PAC that a new PAC should be provisioned in the **PAC Reprovision** field.

   When a user presents a PAC that has less time to live than the reprovision time, a new PAC is provisioned using the existing PAC.

5. Enter the authority identifier that this server sends to users in the **Authority Identifier Info** field.

   The authority identity may be presented by the client software to help select the appropriate PAC by the client.

   - If you enter `auto` in the **Authority Identifier Info** field, the server identifies itself with the name of the server on which Steel-Belted Radius is installed.
   - If you enter any other string in the **Authority Identifier Info** field, the server sends that string as the authority identifier.

**Configuring Inner Authentication Settings**

Inner authentication settings let you specify the manner in which the inner authentication step operates. To configure inner authentication settings for the EAP-FAST protocol:

1. Click the **Inner Authentication** tab in the Edit FAST Authentication method dialog.

2. Optionally, enter the name of a directed realm in the **Directed Realm** field.

   Specifying the name of a directed realm causes the request to be routed based on the methods listed in the directed realm. Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. Optionally, enter the name of a realm selection script in the **Realm Selection Script** field.

   You must license the Steel-Belted Radius scripting module to use realm selection scripts. For information on the Steel-Belted Radius scripting module, refer to the **Steel-Belted Radius Scripting Guide**.
**Configuring Advanced Server Settings**

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-FAST protocol:

1. Click the **Advanced Server Settings** tab in the Edit FAST Authentication method dialog.

2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange, in the **TLS Message Fragment Length** field.
   
   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.
   
   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).
   
   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.
   
   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.

4. Enable the **Return MPPE Keys** check box to specify whether the EAP-FAST module includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.
   
   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.
   
   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.
   
   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.
   
   These cipher suites are documented in RFC 2246, “The TLS Protocol Version 1,” and other TLS-related RFCs and draft RFCs.
   
   Default value is `0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09`. 

---

Chapter 12: Setting Up EAP Authentication Policies
7. Enter the number of seconds a RADIUS client allows a session to persist before asking the client to re-authenticate in the **Session Timeout** field.

   If you enter 0 in the **Session Timeout** field, the Session-Timeout attribute is not generated.

   — The RADIUS client must be configured to process the Session Timeout return attribute so that it can tell the client to reauthenticate after the session timer has expired.

8. Enter the value to return for the Termination-Action attribute sent in an accepted client in the **Termination Action** field.

   If no value is specified, the Termination-Action attribute is not sent.

---

### EAP-PEAP

The EAP-PEAP (Protected EAP) protocol is similar to EAP-TTLS. Unlike EAP-TTLS, which can tunnel any kind of authentication request (such as PAP or CHAP) and extended attributes, PEAP can tunnel only other EAP protocols inside its connection.

EAP-PEAP works in two phases:

- In Phase 1, the client authenticates the server and uses a TLS handshake to create an encrypted tunnel.
- In Phase 2, the server authenticates the user or machine credentials using an EAP authentication protocol. The EAP authentication is protected by the encrypted tunnel created in Phase 1. The authentication type negotiated during Phase 2 can be any valid EAP type, such as GTC (Generic Token Card) or MS-CHAPv2.

Microsoft’s implementation of PEAP and Cisco’s implementation of PEAP supports different methods of client authentication through the TLS tunnel.

- The Microsoft PEAP implementation requires MS-CHAP-V2 for client authentication.
- The Cisco PEAP implementation supports client authentication by EAP-Generic Token, which Cisco uses both for authenticating token cards and for authenticating users against Windows domain/Active Directory accounts.

The Cisco PEAP implementation supports the ability to hide username identities until the TLS encrypted tunnel is established and authentication phase is complete. The Microsoft PEAP implementation sends the username in clear text in Phase 1 of PEAP authentication.

Steel-Belted Radius supports both Microsoft PEAP and Cisco PEAP.
Configuring EAP-PEAP

You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-PEAP authentication method. For information on configuring your server certificate, see “Configuring Server Certificates” on page 145.

To configure EAP-PEAP on a Steel-Belted Radius server:

1. Select Authentication Policies > EAP Methods to open the EAP Methods panel (Figure 68).

   **Figure 68: EAP Methods Panel**

2. Click the Enable check box for the EAP-PEAP authentication method.

3. Select the EAP-PEAP entry and click the Edit button on the toolbar (or double-click the EAP-PEAP entry).

   The Edit PEAP Authentication Method dialog (Figure 69) opens.

   **Figure 69: Edit PEAP Authentication Method dialog**

4. Use the tabs in the Edit PEAP Authentication Method dialog to configure the following settings:
- Request filters
- Response filters
- Session resumption
- Inner authentication
- Advanced server settings

Each configuration task is described separately below.

**Configuring Request Filters**

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.

You must configure filters using the Filters panel before you can associate them with the EAP-PEAP authentication method. For information on configuring filters, refer to Chapter 11, “Setting Up Filters” on page 97.

To configure request filtering for the EAP-PEAP protocol:

1. Click the **Request Filters** tab in the Edit PEAP Authentication method dialog.

2. Optionally, click the **Transfer Outer Attribs to New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests).

   - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   - If this filter is not specified, no attributes from the outer request are transferred to the inner request.

3. Optionally, click the **Transfer Outer Attribs to Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.
This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

If this filter is not specified, the request remains unaltered.

5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

This filter affects only an outer Access-Accept response that is sent back to a network access device. If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.

If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

**Configuring Response Filters**

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-PEAP protocol:

1. Click the **Response Filters** tab in the Edit PEAP Authentication method dialog.

2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.

   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

3. Optionally, click the **Transfer Inner Attribs to Reject** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.
**Configuring Session Resumption**

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

**NOTE:** For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-PEAP protocol:

1. Click the **Session Resumption** tab in the Edit PEAP Authentication method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session Timeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationally cheap.

3. Enter the integer value that you want returned in a Termination-Action attribute in the **Termination Action** field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached.

   If you enter 0 or if you do not specify a value for this attribute, the EAP-PEAP plug-in does not generate such an attribute. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.
Chapter 12: Setting Up EAP Authentication Policies

Configuring Inner Authentication Settings
Inner authentication settings let you specify the manner in which the inner authentication step operates. To configure inner authentication settings for the EAP-PEAP protocol:

1. Click the Inner Authentication tab in the Edit PEAP Authentication method dialog.

2. Optionally, enter the name of a directed realm in the Directed Realm field.

   Specifying the name of a directed realm causes the request to be routed based on the methods listed in the directed realm. Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. Optionally, enter the name of a realm selection script in the Realm Selection Script field.

   You must license the Steel-Belted Radius scripting module to use realm selection scripts. For information on the Steel-Belted Radius scripting module, refer to the Steel-Belted Radius Scripting Guide.

Configuring Advanced Server Settings
Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-PEAP protocol:

1. Click the Advanced Server Settings tab in the Edit PEAP Authentication method dialog.

2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange in the TLS Message Fragment Length field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter the number of seconds in which a transaction must finish in the Max Transaction Time field.

4. Enable the Return MPPE Keys check box to specify whether the EAP-FAST module includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.
You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, “The TLS Protocol Version 1,” and other TLS-related RFCs and draft RFCs.

   Default value is **0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05, 0x04, 0x07, 0x09**.

7. Enter the minimum version of the PEAP protocol that the server should negotiate in the **PEAP Minimum Version** field.

   If you enter 0, the server negotiates version 0, which is compatible with Microsoft’s initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

   If you enter 1, the server negotiates version 1, which is compatible with Cisco’s initial PEAP implementation (shipped in Cisco ACU).

   The value entered in this setting must be less than or equal to the value entered for the **PEAP Maximum Version** field.

8. Enter the maximum version of the PEAP protocol that the server should negotiate in the **PEAP Maximum Version** field.

   If you enter 0, the server negotiates version 0, which is compatible with Microsoft’s initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

   If you enter 1, the server negotiates version 1, which is compatible with Cisco’s initial PEAP implementation (shipped in Cisco ACU).

   The value entered in this setting must be equal to or greater than the value entered for the **PEAP Minimum Version** field.
Configuring Server Certificates

Adding a Server Certificate

To add a certificate to the Steel-Belted Radius server:

1. Select Authentication Policies > Certificates to open the Certificates panel (Figure 68).

Figure 70: Certificates Panel

2. Click the Add button.

3. When the Select Server Certificate dialog (Figure 71) opens, navigate to the location of your server certificate, select the certificate you want to use, and click Open.
Configuring the Server

Depending on your authentication requirements, you may need to configure Steel-Belted Radius to work with an external SQL or LDAP database, RSA SecurID service, or TACACS+

**Configuring External Databases (Solaris/Linux)**

If you run Solaris or Linux and want to use external databases for authentication or accounting purposes (and you did not configure this feature when prompted by the Steel-Belted Radius installation script), you can set up external database configuration settings.
To configure Steel-Belted Radius to work with an external database:

1. Optionally, perform the instructions in Chapter 15, “Configuring SQL Authentication” on page 193 and/or Chapter 16, “Configuring SQL Accounting” on page 207.

2. If you want to use Steel-Belted Radius with an LDAP database, review your LDAP database vendor’s documentation.

3. Perform the instructions in “Configuring LDAP Authentication” on page 222.

**Configuring SecurID Authentication**

If you want to use SecurID authentication, you must configure Steel-Belted Radius to communicate with the RSA SecurID server.

Perform the following steps to configure a Steel-Belted Radius server to work with an RSA SecurID server. If you are not familiar with the RSA SecurID server, contact your RSA SecurID server administrator for assistance.

1. Verify that the Steel-Belted Radius server has an entry on the RSA SecurID server.

   Start the RSA SecurID server administration program and display the list of clients. If the list of clients does not include the Steel-Belted Radius server, select **Client > Add Client** and complete the Client window, giving the Steel-Belted Radius server a Client type of **Net OS Client**.

2. Copy the `sdconf.rec` file from the `\ACE\data` directory on the RSA SecurID server to the appropriate directory on the Steel-Belted Radius server:
   - **Windows**: `C:\winnt\system32`
   - **Solaris/Linux**: the directory that contains the `radius` daemon on the Steel-Belted Radius server.

3. Edit the `[SecurID]` section of `radius.ini`. The `radius.ini` file is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the `CachePasscodes` field is set to `yes` and the `SecondsToCachePasscodes` field is set to an appropriate number of seconds. These settings ensure that authenticated SecurID users can open a second B-channel during an ISDN connection.

4. Edit the `[SecurID]` section of the `eap.ini` file, which is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the EAP settings in this section are enabled (remove the semi-colon from the start of each line) if you plan to use RSA SecurID authentication with EAP Generic-Token protocol support. The client system must support this protocol as well for this combination to work.

5. If you copy the `sdconf.rec` file after the Steel-Belted Radius service (daemon) has been started, or if you edit the `radius.ini` or `eap.ini` files after Steel-Belted Radius has been started, stop and restart Steel-Belted Radius.
6. Verify connectivity between the Steel-Belted Radius server and the RSA SecurID server.

   The RSA SecurID server offers a monitoring window on which it logs every authentication transaction, complete with the reason for the accept or reject decision. You can verify that pass-through to RSA SecurID is working, by creating a SecurID User called <ANY> and then attempting to access the network. Look for your request on the RSA SecurID monitor screen. If access is denied, you'll know that there's a configuration problem. Try these steps again, or contact your RSA SecurID administrator for assistance.

These steps complete initial setup of the two servers. To fully enable pass-through authentication to the RSA SecurID server, you must also set up the SecurID authentication method.

**Configuring the Location of the sdconf.rec File**

The VAR_ACE variable in the sbrd script file (Solaris/Linux) lets you specify the directory holding the sdconf.rec file. The VAR_ACE variable must be exported so that Steel-Belted Radius can use it.

For example:

```bash
VAR_ACE="radiusdir/ace"
export VAR_ACE
```

This variable is set by default in the file to point to the radiusdir directory. If the variable is not set at all in the file, the server sets the value of this variable to /var/ace.

**Configuring TACACS+ Authentication**

If you want to use TACACS+ authentication, you must configure Steel-Belted Radius to communicate with the TACACS+ server.

Perform the following steps to configure a Steel-Belted Radius server to work with a TACACS+ server.

1. Verify the tacplus.ini file is present in the Steel-Belted Radius directory.

   The tacplus.ini file must be present in the same directory as the Steel-Belted Radius service (in the case of Windows, usually C:\Program Files\Juniper Networks\Steel-Belted Radius\Service), or daemon (in the case of Solaris/Linux). This happens automatically following installation.

2. Edit the tacplus.ini file to identify the shared secret and host machine that you use for TACACS+. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

3. If you edit tacplus.ini after Steel-Belted Radius has been started, then you must stop and restart it before your changes take effect.

To enable pass-through authentication to the TACACS+ server, you must also set up the TACACS+ authentication method. For more information, see “Configuring TACACS+ Authentication” on page 148.
Activating EAP Methods

The EAP Methods panel (Figure 74) permits you to activate authentication methods and define the order in which different authentication methods are attempted.

Figure 74: EAP Methods Panel

To use the EAP Methods panel:

2. Click the Enable check box to enable the EAP authentication methods you want Steel-Belted Radius to use.

To revert to the previous settings, click Reset.

Configuring EAP Settings

When Steel-Belted Radius receives a username, it does not know in advance to which authentication category this user belongs. It must try each method that it currently has configured and enabled. The authentication methods list allows you to fine-tune the sequence of authentication attempts.

NOTE: The EAP Setup dialog displays the authentication methods that have been enabled (by editing the Enabled setting in the appropriate *.aut file).

To set up EAP settings for an authentication method:

1. Select the authentication method you want to set up in the Authentication Methods tab.
2. Click the EAP Setup button.

The Setup EAP dialog (Figure 75) opens.
3. Optionally, change the order in which the methods are tried by highlighting a method and clicking the Up or Down buttons.

4. To activate a method, (so that it can be used for authentication), click the Active check box.

   If you want to deactivate a method (so that it is not used for authentication), clear the applicable Active check box.

5. If you want to restrict use of this authentication method to requests that contain EAP credentials, click the Use EAP authentication only check box.

   When this option is enabled, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials, and bypasses the authentication method if an authentication request specifically requests an EAP protocol that is not listed in the authentication method’s EAP-Type list in the eap.ini file.

6. If you want Steel-Belted Radius to use an automatic EAP helper to generate credentials for a user, click the Handle via Auto-EAP first check box.

   You should clear the check box if an authentication method is capable of handling EAP credentials on its own (without an EAP helper).

   Refer to “First-Handle-Via-Auto-EAP Setting” on page 108 for more information.

7. Click Save to return to the Authentication Methods tab.
Configuring Authentication Rejection Messages

When Steel-Belted Radius issues an Access-Reject message in response to a failed authentication request, it can identify the reason why the request was rejected. You can configure the message text returned to the RADIUS client (and possibly to the user, if the RADIUS client forwards the message) when a particular type of error occurs. This text is inserted into the standard RADIUS attribute Reply-Message within the Access-Reject response.

To configure the text for authentication rejection messages:

1. Open the Authentication Policies panel.
2. Click the Reject Messages tab (Figure 76).
3. Use the Unknown User field to specify the message Steel-Belted Radius returns when the username and password authentication failed.
4. Use the Checklist failure field to specify the message Steel-Belted Radius returns when the user was authenticated but is being rejected because the RADIUS request did not fulfill the requirements of the checklist.
5. Use the Invalid attribute field to specify the message Steel-Belted Radius returns when the request contained an attribute in violation of the RADIUS specification.
6. Use the Other field to specify the message Steel-Belted Radius returns when some other error, such as a resource failure, occurred.
7. When you are asked to confirm that you want to save your changes, click Yes.
This chapter describes how to configure and use the centralized configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment.

About Replication

Steel-Belted Radius supports the replication of RADIUS configuration data from a primary server to one or more replica servers within a replication realm. Replication provides administrators with an easy way to configure multiple servers that require the same information. Depending on network configuration, you can use replication to increase AAA capacity, balance AAA traffic across RADIUS servers, or ensure that authentication services are not interrupted if access to a primary or replica server is interrupted (redundancy).

Figure 77 illustrates an environment where RADIUS traffic is load-balanced by configuring each network access device to authenticate users through a different RADIUS server (solid line). If a RADIUS server becomes unavailable, the NAD can fail over to its backup RADIUS server (dotted line).
All the servers within a realm reflect the current configuration specified by the network administrator: the network administrator modifies the configuration on the primary server, and the primary server propagates the new configuration to its replica servers. For example, after a network administrator configures a new RADIUS client or profile on the primary server, the network administrator tells the primary server to publish a date-stamped configuration package file that contains the updated configuration information. After publication, the primary server notifies each replica server that a new configuration package is ready. Each replica server then downloads and installs the configuration package to update its settings.

**Figure 78: Publication and Distribution of Replication Packages**

The primary server maintains a list of the replica servers that have registered with it. The primary server uses this list to track which servers to notify after it publishes an updated configuration package to resynchronize the configuration of replica servers.

---

**NOTE:** You should limit access to the directory in which you store configuration packages on Windows servers to the CREATOR OWNER, SYSTEM, and Administrators. To set file access permissions for the `\Radius\Service` directory, right-click the directory icon, click the **Security** tab, click the **Allow** and **Deny** check boxes to limit access to authorized users.

By default, file permissions for configuration packages on Solaris/Linux servers are set to `rw-rw----`, which excludes users other than the file owner and the owner’s group from displaying the contents of file packages.

If the primary server needs to be taken out of service for an extended period, the network administrator promotes one of the replica servers to be the new primary server. Thereafter, the other replica servers copy the configuration package from the promoted primary server.
The following types of information are included in a replication package.

- Server information
- RADIUS client information
- User information
- Profile information
- Proxy target information
- EAP method configurations
- Filters
- RADIUS tunnel information
- Name parsing information
- Authentication method information
- Authentication realm information
- Rejection messages
- Javascript (.jsi) files

You administer this information by connecting the SBR Administrator to the primary server: the information is propagated to the replica servers in the domain. (If you connect the SBR Administrator to a replica server, you can view this information, but you cannot modify it.)

The following types of information are not included in a replication package:

- Address pool information—You administer address pools for a server by connecting the SBR Administrator to that server. Because an address must not be assigned to two users at the same time, each server in a realm must have its own address pools, and these pools must not overlap.
- Administrator information—Administrator information must be configured for each primary and replica server separately.
- Statistics information—Server statistics are not replicated. You can view statistics for replica servers when you connect SBR Administrator to the primary server.
- Report information—Report information is not replicated. To obtain report information for a primary or replica server, connect SBR Administrator to the applicable server.
- Steel-Belted Radius configuration files—Configuration files (*.ini files (other than filter.ini and eap.ini), *.aut files (other than peapauth.aut, ttlsauth.aut, fastauth.aut, tlsauth.aut, and talsauth.eap), and *.dir files are not replicated. When you change configuration files on the primary server, you must copy the modified files to the appropriate directory on each replica server.

**NOTE:** Configuration packages are retained until they are replaced. An old configuration package is automatically deleted 24 hours after a new configuration package is published.
Replication Requirements

Servers in a replication realm must comply with the following requirements.

- All servers in a replication realm must be running the same operating system (Windows, Solaris, or Linux).
- All servers in a replication realm must be configured to support the same types of users (domain, host, RSA SecurID, TACACS+, or UNIX).
- If RSA SecurID is enabled on the primary server, RSA SecurID must be enabled on the replica servers, and all servers in the realm must have consistent sdconf.rec files.
- The system clocks on servers in a replication realm must by synchronized to within 10 minutes of one another and their time zones must be configured correctly. Steel-Belted Radius uses the system clock value and time zone setting to convert local time to Universal Time Coordinated (also known as Greenwich Mean Time) when evaluating synchronization. If possible, you should use a Network Time Protocol (NTP) server to set the system clock on all servers automatically.
- All servers in a replication realm must use the same TCP port to exchange replication information. The default port for replication communication is TCP 1812, though you can specify another port for replication traffic by modifying the radius.ini file.
- If a firewall stands between servers in a replication realm, the firewall must be configured to pass traffic on the port used for replication communication.

Configuring Replica Servers

The Replication panel (Figure 79) lets you add servers to a replication realm, initiate publication of a replication package by a primary server, and notify replica servers that they should download and install a new replication package.
Configuring Replica Servers

Chapter 13: Configuring Replication

Figure 79: Replication Panel

Adding a Replica Server

In most situations, you add a replica server to a realm as follows:

1. Copy the `replica.ccmpkg` configuration package file from the primary server to a directory on the host you want to add as a replica server.

   Note that the replica.ccmpkg file contains sensitive information, and should not be stored in a publicly accessible location, such as a file server or shared directory.

2. Install the Steel-Belted Radius server software on the host you want to add as a replica server.

3. When the installer (Windows) or configuration script (Solaris/Linux) asks what kind of server you are installing, choose **Replica** and, when prompted, enter the path to the `replica.ccmpkg` file.

4. Restart the host you want to add as a replica server.

   The replica server registers itself with the primary server automatically after it is restarted. Thereafter, the replica server automatically connects to the primary server once an hour to check whether an updated configuration package is available.

In some circumstances, however, you may want to add a replica server to the server list on the primary server manually so that it shows up immediately. To register a replica server manually:

1. Run SBR Administrator and connect to the primary server.

2. Open the Replication panel.

3. Click the **Add** button.

   The Add Server dialog (Figure 80) opens.
4. Enter the name of the RADIUS server in the **Name** field. Although you can assign any name to a RADIUS server, you should use the device's hostname to avoid confusion.

5. Enter the replication secret for the RADIUS server in the **Secret** field. For privacy, asterisks are echoed as you type. You can click the **Unmask** check box to display the characters in the shared secret.

6. Enter one or more IP addresses for your server.
   a. Click the **Add** button.
   b. When the Add IP Address dialog (Figure 81) opens, enter an IP address you want to associate with the server in the **Address** field and click **Add**.

   **Figure 81: Add IP Address Dialog**

   c. Repeat Step 5b until you have finished adding IP addresses for the server.
   d. Click **Close**.

7. Click **OK**.

**Enabling a RADIUS Server**

To enable a RADIUS server:

1. Open the Replication panel.

2. Select the RADIUS server you want to enable and click the **Edit** button (or double-click the RADIUS server entry).
The Edit Server dialog (Figure 82) opens.

**Figure 82: Edit Server Dialog**

3. Click the **Enabled** check box.

4. Click the **Save** button.

**Deleting a RADIUS Server**

To delete a replica server from a realm:

1. Open the Replication panel.

2. Select the RADIUS server entry you want to delete.

3. Click the **Delete** button on the SBR Administrator toolbar.

4. When you are prompted to confirm the deletion request, click **Yes**.

**Publishing Server Configuration Information**

If you change the configuration of your primary server, you must publish the modified configuration so that your replica servers can download the modified settings.

To publish server configuration information:

1. Open the Replication panel.

2. Click the **Publish** button on the toolbar.

This creates a file called `/radius/packages/timestamp_RSA.cmpkg` (Solaris/Linux) or `\Radius\Service\packages\timestamp_RSA.cmpkg` (Windows), where *timestamp* reflects the date and time the package was created.
**Notifying Replica RADIUS Servers**

Under normal circumstances, a replica server connects to its primary server every hour to check whether a new `replica.cmpkg` file has been published. If necessary, a network administrator can manually notify a replica server to download and install the current configuration package from the primary server. Manual notification is useful when network issues prevent the automatic download and installation of a configuration package when it is first published, and the configuration on the replica no longer matches the configuration on the primary server.

To notify replica servers that new configuration information has been published:

1. Open the Replication panel.
2. Select the replica server you want to notify.
3. Click the **Notify** button on the toolbar.

The replica server downloads and installs its configuration package from the primary server. After the package is installed, the replica server is resynchronized with the primary server.

**NOTE:** You can display the Replication panel to determine the status of your replica servers.

---

**Designating a New Primary RADIUS Server**

You can change which server within a realm is designated as the primary server for that realm.

To designate a new primary server:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as `root`.
3. Open a command window and navigate to the `\Radius\Service` directory (Windows) or `/opt/juniper/radius` directory (Solaris/Linux).
4. Run the `sbrsetuptool` utility with the `promote` option.

```
# sbrsetuptool -promote
```

The utility creates a configuration package to change this server to the primary server.

5. Restart the updated replica server to make it the new primary server.
6. Publish a new configuration package administratively to configure all replica servers to use the new primary server.

After you designate a new primary server for a realm, the old primary server becomes a replica server automatically.
**Recovering a Replica After a Failed Download**

If a replica server fails during the download of a configuration package, its configuration may be corrupted or it may have a stale secret.

To recover after a failed download:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as `root`.
3. Open a command window and navigate to the `\Radius\Service` directory (Windows) or `/opt/juniper/radius` directory (Solaris/Linux).
4. Run the `sbrsetuptool` utility with the `identity` option and information on where to download configuration information.

   To obtain configuration from a configuration package, issue the following command:

   ```
   # sbrsetuptool -identity REPLICA -reppkg pathname
   ```

   where `pathname` specifies the path to a `replica.ccmpkg` package.

   To obtain configuration from the primary server for the realm, issue the following command:

   ```
   # sbrsetuptool -identity REPLICA -primary name address secret
   ```

   where `name` specifies the DNS name of the primary server, `address` specifies the IP address of the primary server, and `secret` specifies the shared secret used to authenticate configuration downloads.

5. Restart the updated replica server so that it can load its new configuration.

After the replica server is restarted, it will be re-synchronized with the current primary server.

**Changing the Name or IP Address of a Server**

You may need to change the DNS name or IP address assigned to a primary or replica server if your network changes.

To change the DNS name or IP address of a primary or replica server:

1. Stop the RADIUS service on the RADIUS server you want to change.
2. Log into the RADIUS server as `root`.
3. Open a command window and navigate to the `\Radius\Service` directory (Windows) or `/opt/juniper/radius` directory (Solaris/Linux).
4. Run the `sbrsetuptool` (Solaris/Linux) utility with the `identity` option.

   To rename a primary server, enter the following command:
To rename a replica server, enter the following command:

```
# sbrsetuptool -identity PRIMARY
```

5. Restart the updated server so that it can load its new configuration.

6. Run the SBR Administrator and modify the DNS name or IP address for the server you want to rename. Verify that the secret on the renamed server is correct.

You may need to use the Replication panel to delete the old server name from the list of servers in the realm.

7. Publish the modified configuration to propagate the name change to the replica servers.

---

### Replication Error Messages

The following tables list possible causes for error messages caused by replication issues.

#### Error Messages on Replica Servers

Table 18 lists possible causes for error messages on replica servers in a replication realm.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post Errors (Errors with Notification from Primary)</td>
<td>CRadManagedServerNotifyPost::ExecutePost invalid signature!</td>
<td>Mismatched replication secret.</td>
</tr>
<tr>
<td></td>
<td>CRadManagedServerNotifyPost::ExecutePost invalid sequence number</td>
<td>Two posts have the same sequence number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The clocks on the primary and replica are more then 10 minutes apart.</td>
</tr>
<tr>
<td></td>
<td>CRadManagedServerNotifyPost::ExecutePost decrypt failed</td>
<td>Shared secret failed to decrypt.</td>
</tr>
<tr>
<td></td>
<td>CRadManagedServerNotifyPost::ExecutePost invalid &lt;body&gt; missing parameters</td>
<td>Post had an invalid xml request.</td>
</tr>
<tr>
<td>Update Errors (Errors with Published package from Primary)</td>
<td>CRadManagedServerUpdate::DoStart Failed to open 'file_name' for writing</td>
<td>Temp directory does not exist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Temp directory or file have incorrect permissions.</td>
</tr>
<tr>
<td></td>
<td>CRadManagedServerUpdate::StartUpdates has already started update</td>
<td>Update is already in progress.</td>
</tr>
</tbody>
</table>
**Table 18: Error Messages on Primary Servers** (continued)

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRadProxyPost:: ExecutePost</td>
<td>invalid signature!</td>
<td>Mismatched replication secret.</td>
</tr>
<tr>
<td>CRadProxyPost:: ExecutePost</td>
<td>invalid sequence number</td>
<td>Two posts have the same sequence number. The clocks on the primary and replica are more than 10 minutes apart.</td>
</tr>
<tr>
<td>CRadProxyPost:: ExecutePost</td>
<td>decrypt failed</td>
<td>Shared secret failed to decrypt. Bad Replication Secret secret.</td>
</tr>
</tbody>
</table>
| CRadProxyPost:: ExecutePost | invalid  
\textless body\textgreater  missing parameters | Post had an invalid XML request. |

**Error Messages on Primary Servers**

Table 19 lists possible causes for error messages on primary servers in a replication realm.

**Table 19: Error Messages on Primary Servers**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRadManagedServerUpdate:: DownloadPackage HTTP POST</td>
<td>Error in transmitting request to Primary (timeout during transmit).</td>
<td></td>
</tr>
<tr>
<td>CRadManagedServerUpdate:: DownloadPackage HTTP headers parsing error</td>
<td>Error in receiving package. Typically caused by a timeout during receive resulting from an invalid package.</td>
<td></td>
</tr>
<tr>
<td>CRadManagedServerUpdate:: DownloadPackage connection primary IP Addr error errCode Primary ID</td>
<td>Replica failed to connect with Primary. Primary not running.</td>
<td></td>
</tr>
<tr>
<td>CRadManagedServerUpdate:: DownloadPackage exceeded iterations limit while communicating with CCM</td>
<td>Update failed after three attempts.</td>
<td></td>
</tr>
<tr>
<td>CRadManagedServerUpdate:: ProcessPackage signature mismatch</td>
<td>Secrets on Replica and Primary do not match.</td>
<td></td>
</tr>
<tr>
<td>CRadManagedServerUpdate:: ProcessPackage CCM error: 'Error String' 'Parameter'</td>
<td>Error parsing downloaded packages.</td>
<td></td>
</tr>
</tbody>
</table>
| CRadManagedServerUpdate:: ProcessPackage Failed to open \
\textless file_name \
\textless file_name > for writing | Temp directory does not exist. Temp directory or file has incorrect permissions. |
| CRadManagedServerUpdate:: ProcessPackage thumbprint mismatch | Invalid package. Republish the package. |

**Proxy Errors (Statistics retrieve errors)**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRadProxyPost:: ExecutePost</td>
<td>invalid sequence number</td>
<td>Two posts have the same sequence number. The clocks on the primary and replica are more than 10 minutes apart.</td>
</tr>
<tr>
<td>CRadProxyPost:: ExecutePost</td>
<td>decrypt failed</td>
<td>Shared secret failed to decrypt. Bad Replication Secret secret.</td>
</tr>
</tbody>
</table>
| CRadProxyPost:: ExecutePost | invalid  
\textless body\textgreater  missing parameters | Post had an invalid XML request. |
### Table 19: Error Messages on Primary Servers

<table>
<thead>
<tr>
<th>Notify Target (Both Notify and Publish send a notification)</th>
<th>CRadConfigManagedServerHTTP Notification::NotifyTarget failed to fetch replicaId</th>
<th>Notify failed to communicate with replica. Replica is not running, or check Replica DCF log for more information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication Provider (requests from GUI to Notify or Publish)</td>
<td>CRadConfigPublicationProvider:: UpdateResource notify invoked when not Primary</td>
<td>Attempted to Notify as a Replica, this is only allowed from a Primary.</td>
</tr>
<tr>
<td>Publication Post (parsing of Post from replica to get data)</td>
<td>CRadConfigServerProviderPost:: ExecutePost signature mismatch with server.</td>
<td>Replica and Primary have different replication secrets.</td>
</tr>
<tr>
<td>Proxy Errors (Statistics retrieve errors)</td>
<td>CRadProxyClient:: Send failed to fetch replicaId</td>
<td>This can occur if two administrators are running instances of SBR Administrator, one administrator deletes a replica, then the other administrator tries to publish to that replica.</td>
</tr>
<tr>
<td></td>
<td>CRadProxyClient:: SendData HTTP POST error.</td>
<td>Connection error with replica.</td>
</tr>
</tbody>
</table>
Chapter 14
LDAP Configuration Interface

This chapter describes:

- The file used to enable and configure the LDAP configuration interface (LCI)
- An overview of the LCI and LDAP utilities
- A description of the LDAP virtual schema
- Information about how to use LDAP utilities to configure the Steel-Belted Radius database
- Sample LDIF files that control the execution of LDAP utilities
- Information about how to view rate statistics variables with LDAP utilities

LDAP Configuration Interface File

The radius.ini file establishes settings for the LDAP configuration interface. For more information about radius.ini, refer to the Steel-Belted Radius Reference Guide.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>radius.ini</td>
<td>Specifies whether the LCI is enabled, the port used for LCI communication, and the interfaces on which Steel-Belted Radius listens for LCI requests.</td>
</tr>
</tbody>
</table>

About the LDAP Configuration Interface

NOTE: The LDAP Configuration Interface is an optional add-on for the Enterprise edition of Steel-Belted Radius. You must license the LDAP Configuration Interface before you can configure or use it.

The LDAP Configuration Interface (LCI) provided by Steel-Belted Radius consists of an LDAP interface in the Steel-Belted Radius server and an LDAP virtual schema. The LDAP virtual schema presents the structure of the Steel-Belted Radius database in a manner that can be understood by the LDAP client utilities. The LCI uses the virtual schema to retrieve, modify, and delete entries in the database.
Figure 83 illustrates the relationship between LDAP components, the Administrator, and the configuration database.

**Figure 83: LDAP Components**

![Diagram of LDAP Components](image)

**LDAP Utilities**

Freeware LDAP utilities, such as `ldapsearch`, `ldapdelete`, and `ldapmodify`, act as clients of the LDAP interface. LDAP utilities let you read and modify an LDAP database.

- **Idapsearch**—The `ldapsearch` utility locates and retrieves LDAP directory entries. The `ldapsearch` utility opens a connection to an LDAP interface using the specified distinguished name and password, binds, and locates entries based on the specified search filter. A search can return a single entry, an entry’s immediate subentries, or an entire tree or subtree. Search results are returned in LDAP Data Interchange Format (LDIF) format.

- **Idapdelete**—The `ldapdelete` utility deletes entries from an existing LDAP directory. `ldapdelete` opens a connection to the specified server using the distinguished name and password you provide, binds, and deletes the entry or entries.

- **Idapmodify**—The `ldapmodify` utility adds or modifies entries in an existing LDAP directory. `ldapmodify` opens a connection to an LDAP interface using the distinguished name and password you supply, binds, and adds or modifies the entries based on the LDIF update statements contained in a specified file. Because `ldapmodify` uses LDIF update statements, `ldapmodify` can do everything `ldapdelete` can do.
**LDAP Requests**

LDAP requests are submitted in two ways:

- By specifying options on the LDAP configuration interface command line.
- By placing instructions and data into an LDIF file, which you then process by invoking an LDAP command line utility using the `-f` option.

Because communication between the LDAP client and server is unencrypted, the LDAP utilities should be run on the same computer as Steel-Belted Radius.

**Downloading the LDAP Utilities (Windows)**

To use the LCI, you need the `ldapsearch`, `ldapmodify`, and `ldapdelete` utilities. You can download freeware Windows LDAP utilities as follows:


2. When the Sun ONE Directory SDK (software development kit) download page appears, click the **Download** link at the bottom of the page.

3. If you are prompted to register yourself, complete the registration form.

4. When you are prompted to accept the license agreement, click the **Accept** button and then click **Continue**.

5. Download the SDK by clicking the link for the version of the SDK that is appropriate for your computer.

   Versions of the SDK are available for Solaris, Linux, and Windows.

6. When the download is completed, extract the files from the compressed image to a directory on your computer.

To run the LDAP utilities, execute them from this directory. Note that, if you set the path environment variable to point to this directory, you can run them from any location on the system.

---

**NOTE:** The examples that follow assume you are using the LDAP utilities provided as part of the Sun ONE Directory SDK. If you are using LDAP utilities from another source, the command options you use may be different. Consult the documentation for your LDAP utilities for more information.

**LDAP Version Compliance**

The LDAP interface in Steel-Belted Radius complies with version 2 of the LDAP specification. You should use the `-V 2` command option to direct the utilities to use version 2 features. For example:

```
ldapmodify -c -V 2 -p 354 -D "cn=admin,o=radius" -w radius -f filename
```
Configuring the LDAP TCP Port

To avoid conflicts with LDAP services that may already be installed, the default port number for communication between Steel-Belted Radius and the LDAP client is 667. You can configure Steel-Belted Radius to use a different TCP port to communicate with an LDAP client. For example, you can change this port number to 389, the standard LDAP TCP port, if you are certain doing so will not create port number conflicts with other applications.

The following example configures Steel-Belted Radius to use TCP port 354.

1. In the radius.ini file, uncomment the [LDAP] section, set Enable to 1, and set the TCPPort field to the port number you want to use. For example:

   ![radius.ini example](example_radius.ini)

   You must specify the port number (by means of the -p option) when you run the LDAP utilities. For example:

   ```
   ldapsearch -V 2 -p 354 -D "cn=admin,o=radius" -w radius -s sub -T -b "radiusclass=Client,o=radius" radiusname=*
   ```

Configuring the LCI Password

After you enable the LCI, you should change the default LCI password to prevent unauthorized LDAP clients from accessing your database. After you install the LDAP utilities and verify that they work, perform the following steps:

1. Create a text file called temp.ldif with the following contents:

   ```
   dn: radiusclass=server,o=radius
   changetype: modify
   replace: server-password
   server-password: new-password
   ```

   where new-password is the LCI password you want to use.

2. Change the radius.ini [LDAP] setting to Enable=1.

3. Restart Steel-Belted Radius.

4. Execute the following command:

   ```
   ldapmodify -V 2 -h ip-address -p port -D "cn=admin,o=radius" -w oldpassword -f temp.ldif
   ```

   where:

   -h ip-address specifies the IP address of the Steel-Belted Radius server.
   -p port specifies the port number specified in the [LDAP] section of the radius.ini file.
   -w oldpassword specifies the current password (which is radius by default).
5. Verify that the password change was successful by executing the following command:

```
ldapsearch -V 2 -h ip-address -p port -D "cn=admin,o=radius" -w newpassword -s sub -T -b "o=radius" radiusclass=server
```

where:

- `-h ip-address` specifies the IP address of the Steel-Belted Radius server.
- `-p port` specifies the port number specified in the [LDAP] section of the `radius.ini` file.
- `-w newpassword` specifies the password configured in the `temp.ldif` file.

After you verify that the password change has been successful, delete the `temp.ldif` file and any other file that contains a cleartext copy of the modified LCI password.

The LDAP Configuration Interface does not support Secure Sockets Layer (SSL).

---

**LDAP Virtual Schema**

The LDAP interface uses the virtual schema (illustrated in Figures 84–88) to represent the structure of the Steel-Belted Radius database. LDAP clients use the virtual schema to exchange configuration data over the LDAP configuration interface.

**NOTE:** Your edition of Steel-Belted Radius may not support all branches of the schema illustrated in Figures 84 through 88.

Many of the top-level items in the LDAP virtual schema correspond to windows and panels in the SBR Administrator.

**Table 21: LDAP Schema and SBR Administrator Dialogs**

<table>
<thead>
<tr>
<th>Item</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>radiusclass=client</td>
<td>Chapter 4, “Administering RADIUS Clients” on page 41</td>
</tr>
<tr>
<td>radiusclass=native-user, securid-user, ...</td>
<td>Chapter 5, “Administering Users” on page 47</td>
</tr>
<tr>
<td>radiusclass=profile</td>
<td>Chapter 6, “Administering Profiles” on page 67</td>
</tr>
<tr>
<td>radiusclass=proxy</td>
<td>Chapter 7, “Administering Proxy RADIUS” on page 71</td>
</tr>
<tr>
<td>radiusclass=tunnel</td>
<td>Chapter 8, “Administering RADIUS Tunnels” on page 77</td>
</tr>
<tr>
<td>radiusclass=server</td>
<td>Chapter 12, “Setting Up EAP Authentication Policies” on page 105</td>
</tr>
<tr>
<td>radiusclass=ip-addr-pool</td>
<td>“Setting Up IP Address Pools” on page 85</td>
</tr>
<tr>
<td>radiusclass=ipx-addr-pool</td>
<td>“Setting Up IPX Address Pools” on page 88</td>
</tr>
</tbody>
</table>
Table 21: LDAP Schema (continued) and SBR Administrator Dialogs (continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>radiusstatus=statistics</td>
<td>Chapter 18, “Displaying Statistics” on page 231</td>
</tr>
<tr>
<td>radiusstatus=sessions</td>
<td>“Displaying the Current Sessions List” on page 239</td>
</tr>
</tbody>
</table>

**NOTE:** LDAP searches that call `radiusstatus=sessions` can adversely affect Steel-Belted Radius performance. When possible, you should search using the `sessions_by` keywords.

**NOTE:** `radiusstatus` items can be read, but they cannot be modified.

Figure 84: LDAP Schema (Slide 1 of 5)
Available Attributes:
- Shared-Secret <string>
- Acct-Shared-Secret <string>
- Non-Default-Acct-Port <number>
- IP-Address nnn.nnn.nnn.nnn
- Non-Default-Auth-Port <number>
- Retry-Timeout <milliseconds>
- Retry-Count <number>
- Include-In-Auth-List yes|no
- Accounting forward|local|both

Available Attributes:
- Server-Password <string>
- Server-Password-Enabled 0|1
- Default-Reject-Msg <string>
- Unknown-User-Msg <string>
- Lists-Mismatch-Msg <string>
- Invalid-Lists-Msg <string>
- Auth-Methods <meth1>; <meth2>; ...
- Log-Max-Days <number>
- Tunnel-Delimiter <string>
- Tunnel-Type none|prefix|suffix

Available Attributes (multiple):
- Description <string>
- Range nnn.nnn.nnn.nnn:mm
- Total <number>
- Available <number>
Figure 86: LDAP Schema (Slide 3 of 5)

Available Attributes:
- client <string>
- acct-session-id <number>
- nas-ip-address <string>
- nas-port <string>
- nas-port-type <string>
- acct-multi-session-id <number>
- ipaddressfrompool <string>
- ipxaddressfrompool <string>
- framed-ip-address <string>
- session-start-time <time>
- fullname <string>
- username <string>
- tribe <string>
- called-station-id <string>
- calling-station-id <string>
- elapsed <number>

* NOTE: LDAP searches that call
radiusstatus=sessions can adversely affect
Steel-Belted Radius performance. When
possible, you should search using the
sessions_by keywords.
NOTE: The Enterprise edition of Steel-Belted Radius with the optional LCI add-on does not support the Statistics items.
LDAP Rules and Limitations

While the LDAP virtual schema diagram shows as much of the detail of the LDAP virtual schema as possible, the following rules and limitations should be considered.

- **Bind request**—All attempts to perform operations on the virtual schema must be preceded by an LDAP Bind request that authenticates the administrator to the Steel-Belted Radius server. The Bind request must reference a Steel-Belted Radius administrative account and must provide the password that authenticates that account. This translates into the following command line options for each invocation of the LDAP utilities:

  \[-D "cn=AdminName,o=radius" -w AdminPassword\]

  where *AdminName* is the administrative account name and *AdminPassword* is its password.

- **Uppercase and lowercase**—The uppercase/lowercase rules for object names are the same as in the SBR Administrator; that is, almost all object names are stored in the database in uppercase format. The exception to this rule is that UNIX User/Group, SecurID User/Prefix/Suffix and TACACS + User/Prefix/Suffix names are maintained in the case specified in the LDIF files.

- **Attributes**—When you enter attributes, make sure that the attribute name matches the name found in the dictionary and that the attribute’s value is consistent with the syntax type for the attribute. Note that the LDAP virtual schema does not list all the dictionary attributes that are available in Steel-Belted Radius.
- **IP addresses**—The `ipaddr-pool` type in the dictionary can represent an IP address or a pool name. If the value specified begins with the marker string `[pool]`, the token that follows the marker string is assumed to be an IP pool name; otherwise, it must be a valid IP address. If it is neither, the operation fails.

  Address ranges in IP address pool objects are specified in the form `IPAddress:NumberOfAddresses`. An example of a valid range is `128.22.12.45:34`.

- **IPX addresses**—The `ipxaddr-pool` syntax type in the dictionary lets you enter an IPX network address (up to 8 hexadecimal digits using the format `0xhhhhhhhh`) or choose a pool name. If the value specified begins with the string `[pool]`, the token that follows the marker is assumed to be an IPX pool name; otherwise, it must be a valid IPX address. If it is neither, the operation fails.

  Address ranges in IPX address pool objects are specified in the form `IPXNetAddress:NumberOfAddresses`. An example of a valid range is `0xa020443b:34`.

- **Substrings**—Some attribute may have a value that consists of a list of strings. For example, the DNIS list in a tunnel entry and the authentication method list may consist of multiple substrings. The rule for specifying the data portion for these lists is that semicolons must delimit substrings. For example, a DNIS list for a tunnel entry might be specified as `555-1212;5551212`.

  If a semicolon needs to appear inside a substring, it must be escaped by placing a backslash character (\) before it.

- **Hexadecimal values**—Hexadecimal numbers (for attributes of syntax type `hex1`, `hex2`, or `hex4`) require a `0x` prefix before the hexadecimal digits; for example, `0x0000149a`.

- **Password syntax**—Passwords that are retrieved from the database may consist of one of the following:
  - A clear-text password of the form `{x-clear}clear-text-password-string` if the password is weakly encrypted in the database.
  - A string of the form `{x-md5}xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx` if the password is stored as a one-way md5 hash.
  - A string of the form `{x-md5}[encrypt]clear-text-password-string` indicates that, although the password is specified in clear-text form, it is to be stored as a hash.

  White space in a password is treated as follows:
  - When clear-text passwords are specified, the password is assumed to begin immediately after the right brace or right bracket. Adding a white space character, such as a space or tab, after the right brace or right bracket causes the white space to be considered part of the password.
  - White space entered at the beginning of the attribute (before the left brace or left bracket) is ignored.
- White space entered between the right brace of `{x-md5}` and the left bracket of `[encrypt]` is ignored.

- All white space specified in the hexadecimal sequence describing a password hash is ignored.

- **Profiles, checklists, and return lists**—Steel-Belted Radius supports user definitions that include attribute subtractions of profile entries. To specify that a user attribute is to be considered a subtraction of a profile attribute, preface the attribute value with the string `%subtract%`.

Steel-Belted Radius permits user and profile checklists to include default values for attributes. Configuring a default value for an attribute means that, if a RADIUS request does not include this attribute, the request should not be rejected. Instead, the value supplied as the default should be used as if it were received as part of the request. To specify that a checklist attribute is to be considered a default attribute, preface the attribute value with the string `%default%`.

Steel-Belted Radius permits user and profile return lists to include attributes whose values are set by copying the contents of received attributes. This feature is referred to as “attribute echoing.” To specify that a return list attribute is to be treated as an echo attribute, enter `%echo%` for the attribute value.

- **Unspecified or 0.0.0.0 RAS IP address**—When you display `acct_stats_by_nasipaddr` information, any RAS entries with an unspecified IP address or an IP address of 0.0.0.0 are omitted. Similarly, when you display `acct_stats_by_nas` information, any RAS entries with an unspecified IP address or an IP address of 0.0.0.0 will have their `nasipaddr` attribute omitted.

- **Duplicate RAS IP addresses**—When displaying `acct_stats_by_nasipaddr` information, two RAS entries that contain the same (non-zero) IP address cause information about one of the entries to be displayed twice. This is the result of the ambiguity of the query and is not a bug.

- **RADIUS client information displayed after deletion**—If you define a RADIUS client entry, send some accounting traffic to it, and then delete the entry, the output of `ldapsearch` queries will continue to list the deleted RADIUS client so that the per-RAS statistics add up to the total RAS statistics.

---

**LDAP Command Examples**

This section explains how to use the `ldapdelete`, `ldapmodify`, and `ldapsearch` utilities to configure the server.

**Searching for Records**

You can use the `ldapsearch` command to extract information from the LDAP tree. The command shown in Figure 89 lets you extract information about all RADIUS Native Users.
Figure 89: ldapsearch Command

```bash
ldapsearch -V 2 -p 354 -h 192.168.45.12
-D "cn=oper,o=radius" -w radadmin -s sub -T -b
"radiusclass=Native-User,o=radius" radiusname=*  
```

Note there must be a blank space between each option (for example, `-p`) and its value (for example, 354). Command syntax is case-sensitive.

Table 22: Searching for Records Using the ldapsearch Command

<table>
<thead>
<tr>
<th>ldapsearch Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-V 2</code></td>
<td>Use LDAP Version 2 to communicate with the server. This option is not required, but it improves the performance of the transaction.</td>
</tr>
<tr>
<td><code>-p 354</code></td>
<td>Use TCP port 354 to communicate with the LDAP interface of the server. The <code>-p</code> value must match the TCPPort setting in the [LDAP] section of <code>radius.ini</code>. If the <code>-p</code> option is not specified, the LDAP utilities contact Steel-Belted Radius on the default port number (TCP port 389).</td>
</tr>
<tr>
<td><code>-h 192.168.45.12</code></td>
<td>Contact a remote host at the specified address or name. By default, <code>ldapsearch</code> tries to connect to the local host.</td>
</tr>
<tr>
<td><code>-D &quot;cn=oper,o=radius&quot;</code></td>
<td>Use the <code>oper</code> administrative account to authenticate the command. <strong>NOTE:</strong> You can use any administrative account name in place of <code>oper</code> in this example. Do not change the <code>o=radius</code> argument.</td>
</tr>
<tr>
<td><code>-w radadmin</code></td>
<td>Use an authentication password of <code>radadmin</code>. <strong>NOTE:</strong> The <code>-w</code> parameter value (in this case, <code>radadmin</code>) must match the password of the account named by the <code>-D</code> parameter.</td>
</tr>
<tr>
<td><code>-s sub</code></td>
<td>Perform a recursive subtree search from the base.</td>
</tr>
<tr>
<td><code>-T</code></td>
<td>Do not wrap long output lines to the next line.</td>
</tr>
<tr>
<td><code>-b &quot;radiusclass=Client,o=radius&quot;</code></td>
<td>Specifies the base from which the search operation starts.</td>
</tr>
<tr>
<td><code>radiusname=*</code></td>
<td>Specifies the selection criteria for the search.</td>
</tr>
</tbody>
</table>

Executing the `ldapsearch` command shown in Figure 89 against a Steel-Belted Radius server containing two Native User definitions would produce an LDIF file similar to the output shown in Figure 90.
Modifying Records

You can use the ldapmodify utility to update the Steel-Belted Radius configuration.

```bash
ldapmodify -c -V 2 -h example.host.com -p 354 -D "cn=oper,o=radius" -w radadmin -f filename
```

Note there must be a blank space between each option (for example, `-p`) and its value (for example, `354`). Command syntax is case-sensitive.

Table 23: Modifying Records Using the ldapmodify Command

<table>
<thead>
<tr>
<th>ldapmodify Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c</code></td>
<td>Run the command in continuous mode; do not stop on errors.</td>
</tr>
<tr>
<td><code>-V 2</code></td>
<td>Use LDAP Version 2 to communicate with the server. This option is not required, but it improves the performance of the transaction.</td>
</tr>
<tr>
<td><code>-h example.host.com</code></td>
<td>Contact a remote host at the specified address or name. If the <code>-h</code> option is not used, ldapsearch connects to the local database.</td>
</tr>
<tr>
<td><code>-p 354</code></td>
<td>Use TCP port 354 to communicate with the LDAP interface of the server. The <code>-p</code> value must match the TCPPort setting in the [LDAP] section of radius.ini. If the <code>-p</code> option is not specified, the LDAP utilities contact Steel-Belted Radius on the default port number (TCP port 389).</td>
</tr>
<tr>
<td><code>-D &quot;cn=oper,o=radius&quot;</code></td>
<td>Use the oper administrative account to authenticate the command. NOTE: You can use any administrative account name in place of <code>oper</code> in this example. Do not change the <code>o=radius</code> argument.</td>
</tr>
</tbody>
</table>
The LDIF files generated by `ldapsearch` differ from those required for input to `ldapmodify`. The `ldapmodify` input files must contain a `changetype` entry immediately after each `dn` entry. The `changetype` entry specifies how to use the data to change the LDAP database.

The full syntax for `changetype` within each transaction is as follows:

```

dn: distinguished-name-of-entry
changetype: keyword
subkeyword: attribute
attribute: value
changetype: keyword
subkeyword: attribute
attribute: value
```

where:

- `keyword` can be `add`, `modify`, or `delete`.
- `subkeyword` can be (respectively): `add`, `replace`, or `delete`.
- `attribute` can be any LDAP attribute in the entry.
- `value` is the value to assign to the attribute.

Repeated `changetype: keyword` entries are not required within a transaction unless you change the keyword. From top to bottom within the transaction, the latest keyword applies until another `changetype: keyword` entry is provided. The following syntax is valid if the same keyword applies throughout the transaction:

```

dn: distinguished-name-of-entry
changetype: keyword
subkeyword: attribute
attribute: value
subkeyword: attribute
attribute: value
subkeyword: attribute
attribute: value
```

`subkeyword: attribute` entries are optional and indicate that you want to apply the change to a specific attribute within the entry. If no `subkeyword: attribute` entries in the transaction are found, the change applies to the entire entry. For example, it is faster to delete an entire entry:

```
dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
```
but if you want to delete only a few attributes from the entry, you can do so:

```
  dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
  changetype: delete
  delete: retry-count
  delete: include-in-auth-list
```

If the subkeyword is `add` or `replace`, an attribute: value entry must appear immediately following the `subkeyword: attribute` entry. If the subkeyword is `delete`, the attribute: value entry does not apply and should be omitted.

The following sample LDIF file could be used with an `ldapmodify` command.

**Figure 91: Sample LDIF File**

<table>
<thead>
<tr>
<th>DN: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>changetype: add</td>
</tr>
<tr>
<td>radiusname: BIGCO.COM</td>
</tr>
<tr>
<td>ip-address: 194.132.5.89</td>
</tr>
<tr>
<td>accounting: both</td>
</tr>
<tr>
<td>retry-count: 3</td>
</tr>
<tr>
<td>retry-timeout: 5000</td>
</tr>
<tr>
<td>shared-secret: testing123</td>
</tr>
<tr>
<td>include-in-auth-list: no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DN: radiusname=BIGGERCO.COM,radiusclass=Proxy,o=radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>changetype: modify</td>
</tr>
<tr>
<td>replace: shared-secret</td>
</tr>
<tr>
<td>shared-secret: hereisthesecret</td>
</tr>
<tr>
<td>replace: ip-address</td>
</tr>
<tr>
<td>ip-address: 192.7.2.121</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DN: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>changetype: modify</td>
</tr>
<tr>
<td>delete: include-in-auth-list</td>
</tr>
</tbody>
</table>

**NOTE:** To delete the proxy entry for TINYCO.COM, issue the following command:

```
  dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
  changetype: delete
```

**Importing Records From Another LDAP Database**

To import entries from one LDAP database into another, run the `ldapsearch` command on the first database. Request only the attributes you want for the new database. When `ldapsearch` completes processing, edit the output LDIF file. After each line that begins with `dn:`, add a single line containing the text `changetype: add`. Once your editing is complete, run an `ldapmodify -f` command that references the new LDIF file. After the `ldapmodify` command is executed, your new database is populated with the records you extracted from the old database.
Chapter 14: LDAP Configuration Interface

The LDIF file shown in Figure 92 is derived from the output of the ldapsearch command. When specified as the input to an ldapmodify -f command, the contents of the file are added to the target database.

Figure 92: Adding Records With an LDIF File

```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
   changetype: add
   objectclass: top
   objectclass: Native-User
   objectclass: user
   radiusname: KEVIN
   password: {x-clear}secret1
   profile: ISDN
   login-limit: 2

dn: radiusname=MICHAEL,radiusclass=Native-User,o=radius
   changetype: add
   objectclass: top
   objectclass: Native-User
   objectclass: user
   radiusname: MICHAEL
   password: {x-clear}secret99
   profile: ISDN
   login-limit: 2
```

Deleting Records

The ldapdelete command allows you to remove records from the LDAP database. For example, to delete entries names USER1 through USER5, add the information shown in Figure 93 to a file called deletedemo.ldf.

Figure 93: Deleting Records With an LDIF File

```
radiusname=USER1,radiusclass=Native-User,o=radius
radiusname=USER2,radiusclass=Native-User,o=radius
radiusname=USER3,radiusclass=Native-User,o=radius
radiusname=USER4,radiusclass=Native-User,o=radius
radiusname=USER5,radiusclass=Native-User,o=radius
```

Now, pass the deletedemo.ldf file to the ldapdelete command.

```
ldapdelete -V2 -h hostname -p 667
   -D "cn=admin,o=radius" -w password -f deletedemo.ldf
```

NOTE: Verify that the dn: values that usually appear in these entries are not a part of the entries in your file, because this will cause the command to fail.

You can use ldapdelete to remove records from the LDAP database without having to supply a file. For example, to delete the native user record identified as USER1, you would enter the following:

```
ldapdelete -V2 -h hostname -p 667
   -D "cn=admin,o=radius" -w password
   "radiusname=USER1,radiusclass=native-user,o=radius"
```
You can cause records to be deleted by means of the `ldapmodify` command, if the entries in the text file contain the line `changentype: delete`. Consider the sample LDIF file named `deletemodify.ldf` shown in Figure 94.

**Figure 94: deletemodify.ldf Example**

```
dn: radiusname=barry,radiusclass=Native-User,o=radius
  changetype: delete

dn: radiusname=maurice,radiusclass=Native-User,o=radius
  changetype: delete

dn: radiusname=robin,radiusclass=Native-User,o=radius
  changetype: delete
```

The `deletemodify.ldf` file can be passed to the `ldapmodify` command as follows:

```
ldapmodify -V2 -h hostname -p 667 -D"cn=admin,o=radius" -w password -f deletemodify.ldf
```

**NOTE:** On some LDAP servers, an error could cause the deletion of a container without prompting for confirmation. This could, in turn, cause the entire directory server to fail.

### LDIF File Examples

This section explains how to construct LDIF files that, when input to the `ldapmodify` command, add entries to the Steel-Belted Radius database.

#### Adding RADIUS Clients with LDIF

The sample LDIF entry shown in Figure 95 adds a RADIUS client named ANNEX105 to the Steel-Belted Radius database.

**Figure 95: Adding RADIUS Clients**

```
dn: radiusname=ANNEX105,radiusclass=Client,o=radius
  changetype: add
  objectclass: top
  objectclass: Client
  radiusname: ANNEX105
  ip-address: 193.162.45.12
  product: Nortel Networks Remote Annex
  shared-secret: testing123
```

The syntax in this LDIF entry is shown in Figure 96.
Adding Users with LDIF

The sample LDIF entry shown in Figure 97 adds a Local (Native) User named KEVIN to the Steel-Belted Radius database.

Figure 97: Adding Users

```ldif
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2
```

The syntax in this LDIF entry is shown in Figure 98.

Figure 98: LDIF Syntax

```ldif
dn: radiusname=String,radiusclass=Client,o=radius
changetype: add
objectclass: top
objectclass: Client
radiusname: String
ip-address: IPAddressOfTheClientDevice
product: Make&ModelChoiceFromVendor.IniFile | ...
shared-secret: SharedSecretThatWasConfiguredOnTheClientDevice
RASClientField: RASClientFieldValue
RASClientField: RASClientFieldValue
```

The LDIF file shown in Figure 99 add a local (native) user named CHRISTIAN, who has various attribute/value pairs assigned to his checklist and return list.
Figure 99: Adding a Native User

```
dn: radiusname=christian,radiusclass=native-user,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: CHRISTIAN
password: {x-clear}password
login-limit: 2

dn: radiuslist=check,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: check
radiuslist: check
NAS-IP-Address: 50.50.50.50
Framed-protocol: PPP

dn: radiuslist=reply,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: reply
radiuslist: reply
framed-ip-address: 100.100.100.100
framed-IP-Netmask: 255.255.255.224
```

Checklists and return lists are objects in the LDAP virtual schema, but the individual RADIUS attributes are not. Therefore, you must use a separate LDIF entry for each checklist and return list object, but each LDIF entry can name multiple attribute/value pairs.

To indicate that a transaction applies to the user’s checklist (rather than to the user entry itself), use the keyword `check` as the value for `radiuslist` and `objectclass` within the transaction. You must assign this value to `radiuslist` in the distinguished name, and again before the list of attributes. You must also assign the value to `objectclass`, above the second `radiuslist` entry.

To indicate the return list, use the keyword `reply`.

The LDIF syntax to add a user entry, complete with a checklist and return list, is shown in Figure 100. Note that the `radiusname` and `radiusclass` values for all of the transactions that apply to the same User entry must be the same.
Adding Proxy Targets with LDIF

The sample LDIF entry shown in Figure 101 adds the proxy RADIUS target BIGCO.COM to the Steel-Belted Radius database.

Figure 101: Adding Proxy Targets

```
dn: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius
changeType: add
objectclass: top
objectclass: Proxy
radiusname: BIGCO.COM
ip-address: 194.132.5.89
accounting: both
retry-count: 3
retry-timeout: 5000
shared-secret: testing123
include-in-auth-list: no
```

The syntax in this LDIF entry is shown in Figure 102.
Figure 102: LDIF Syntax

```
dn: radiusname=StringToParseAsProxyName, radiusclass=Proxy, o=radius
change-type: add
objectclass: top
objectclass: Proxy
radiusname: StringToParseAsProxyName
ip-address: IPAddressOfTheTargetServer
accounting: Both | ...
retry-count: Integer
retry-timeout: Integer
shared-secret: SharedSecretThatWasConfiguredOnTheTargetServer
include-in-auth-list: Yes | No
ProxyField: ProxyFieldValue
ProxyField: ProxyFieldValue
```

Adding Tunnels with LDIF

The sample LDIF entry shown in Figure 103 adds the tunnel ACME.COM to the Steel-Belted Radius database.

Figure 103: Adding Tunnels

```
dn: radiusname=ACME.COM, radiusclass=Tunnel, o=radius
change-type: add
objectclass: top
objectclass: Tunnel
radiusname: ACME.COM
dnis-list: 8005551212;6171231234;12343210
description: Tunnel configuration for Acme Corp.
usage-limit: 24
```

The syntax in this LDIF entry is shown in Figure 104.

Figure 104: LDIF Syntax

```
dn: radiusname=StringToParseAsTunnelName, radiusclass=Tunnel, o=radius
change-type: add
objectclass: top
objectclass: Tunnel
radiusname: StringToParseAsTunnelName
dnis-list: PhoneNumber;PhoneNumber;etc
description: StringDescribingTunnel
usage-limit: IntegerGivingConcurrentConnectionLimit
TunnelField: TunnelFieldValue
TunnelField: TunnelFieldValue
```

186  ■  LDIF File Examples
Adding IP Address Pools with LDIF

The sample LDIF entry shown in Figure 105 adds an IP address pool named POOL1 to the Steel-Belted Radius database.

**Figure 105: Adding IP Address Pools**

```
dn: radiusname=POOL1,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: POOL1
description: Address pool for common users
range: 198.187.100.1:50
range: 198.187.101.1:50
```

The syntax in this LDIF entry is shown in Figure 106.

**Figure 106: LDIF Syntax**

```
dn: radiusname=String,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPAddress:Range
range: IPAddress:Range
```

Adding IPX Address Pools with LDIF

The sample LDIF entry shown in Figure 107 adds an IPX address pool named NETWARE1 to the Steel-Belted Radius database.

**Figure 107: Adding IPX Address Pools**

```
dn: radiusname=NETWARE1,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: NETWARE1
description: IPX network numbers for dial in users
range: 0xffff0a00:500
```

The syntax in this LDIF entry is shown in Figure 108. You may provide multiple IPX address ranges using the `range` field.
Configuring a RADIUS Server with LDIF

The sample LDIF entry shown in Figure 109 lets you configure your Steel-Belted Radius server by adding the Native User authentication method and defining conventions for tunnel name parsing.

Figure 109: Adding a RADIUS Server

```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User
tunnel-delimiter: $
tunnel-type: prefix
```

The syntax in this LDIF entry is shown in Figure 110.

Figure 110: LDIF Syntax

```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User | Solaris User | SecurID Prefix | ...
tunnel-delimiter: Character
tunnel-type: Prefix | Suffix | Neither
ConfigurationField: ConfigurationFieldValue
ConfigurationField: ConfigurationFieldValue
```

Figure 108: LDIF Syntax

```
dn: radiusname=String,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPXAddress:Range
range: IPXAddress:Range
```
Statistics Variables

Server statistics counters record the number of certain types of events. The LCI allows you to read these statistics to monitor the performance of your Steel-Belted Radius server.

**NOTE:** The Enterprise Edition of Steel-Belted Radius with the optional LCI add-on does not support the Statistics items.

**Counter Statistics**

The statistics counters can be accessed via the LCI by executing the following one line command:

```
ldapsearch -V v 2 -h 127.0.0.1 -p 667 -D "cn=admin,o=radius" -w radius -s sub -T -b "radiusstatus=statistics,o=radius" stattype=typeofstatus
```

The following sections illustrate the variables displayed for different settings of the stattype parameter.

**stattype: server**

```
dn: stattype=server,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: server
start-time: 2002/05/08 13:29:08
up-time: 26188
ip-address: 192.168.21.142
version: v 2.20.33
authentication-threads: 0
accounting-threads: 0
total-threads: 0
max-auth-threads: 100
max-acct-threads: 100
max-total-threads: 200
high-auth-threads: 2
high-acct-threads: 0
high-total-threads: 2
```
insufficient-resources: 0
proxy-failure: 0
rejected-by-proxy: 0
transactions-retried: 0
total-retry-packets: 0

**stattype: accounting**

dn: stattype=accounting,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: accounting
start: 0
stop: 0
on: 0
off: 0
total-transactions: 0
invalid-request: 0
invalid-client: 0
invalid-shared-secret: 0
insufficient-resources: 0
proxy-failure: 0
transactions-retried: 0
total-retry-packets: 0

**stattype: proxy**

dn: stattype=proxy,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: proxy
authentication: 0
accounting: 0
total-transactions: 0
timed-out: 0
invalid-response: 0
invalid-shared-secret: 0
insufficient-resources: 0
transactions-retried: 0
total-retry-packets: 0

**Rate Statistics**

Rate statistics are derived from existing counter statistics by taking time into consideration. Rate values calculated for each of these counter statistics consist of the following:

- **Current rate**—The rate measured over the most recent rate interval.
- **Average rate**—The rate measured since the Steel-Belted Radius server was started or since the last time statistics were reset to zero.
- **Peak rate**—The highest rate observed since the Steel-Belted Radius server was started or since the last time statistics were reset to zero.
Additionally, there is a (read-only) time value used in calculations:

- Rate statistics seconds-per interval—The duration (in seconds) of the interval over which the rate statistics are gathered.

To read rate statistics from the LCI, you must set `stattype: rate`. This results in output such as the following:

```plaintext
rate-statistics-seconds-per-interval: 1
auth-request-current-rate: 0
auth-request-average-rate: 0
auth-request-peak-rate: 7
auth-accept-current-rate: 0
auth-accept-average-rate: 0
auth-accept-peak-rate: 1
auth-reject-current-rate: 0
auth-reject-average-rate: 0
auth-reject-peak-rate: 0
acct-start-current-rate: 0
acct-start-average-rate: 0
acct-start-peak-rate: 0
acct-stop-current-rate: 0
acct-stop-average-rate: 0
acct-stop-peak-rate: 0
proxy-auth-request-current-rate: 0
proxy-auth-request-average-rate: 0
proxy-auth-request-peak-rate: 0
proxy-acct-request-current-rate: 0
proxy-acct-request-average-rate: 0
proxy-acct-request-peak-rate: 0
proxy-fail-timeout-current-rate: 0
proxy-fail-timeout-average-rate: 0
proxy-fail-timeout-peak-rate: 0
proxy-fail-badresp-current-rate: 0
proxy-fail-badresp-average-rate: 0
proxy-fail-badresp-peak-rate: 0
proxy-fail-badsecret-current-rate: 0
proxy-fail-badsecret-average-rate: 0
proxy-fail-badsecret-peak-rate: 0
proxy-fail-missingresr-current-rate: 0
proxy-fail-missingresr-average-rate: 0
proxy-fail-missingresr-peak-rate: 0
proxy-retries-current-rate: 0
proxy-retries-average-rate: 0
proxy-retries-peak-rate: 0
proxy-auth-rej-proxy-current-rate: 0
proxy-auth-rej-proxy-average-rate: 0
proxy-auth-rej-proxy-peak-rate: 0
proxy-acct-fail-proxy-current-rate: 0
proxy-acct-fail-proxy-average-rate: 0
proxy-acct-fail-proxy-peak-rate: 0
proxy-auth-rej-proxy-error-current-rate: 0
proxy-auth-rej-proxy-error-average-rate: 0
proxy-auth-rej-proxy-error-peak-rate: 0
```
Chapter 15
Configuring SQL Authentication

This chapter presents an overview of SQL authentication and describes how to configure SQL authentication in Steel-Belted Radius.

About SQL Authentication

Steel-Belted Radius can authenticate against records stored in an external SQL database. Any attribute or set of attributes, such as username and password, can be used to query the database.

External database authentication is typically used when an organization already has a large amount of user information stored in a SQL database, and this information is to be used to authenticate these users using RADIUS. Authentication against an existing database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database.

Steel-Belted Radius offers the SQL authentication feature as a plug-in software module. Key features of the SQL plug-in include:

- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- The SQL statement supports a wide range of arithmetic and string expressions as part of the statement.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- Multiple authentications may be overlapped at the same time.
- The SQL authentication method, which appears in the Authentication Policies panel in SBR Administrator, can be activated/deactivated and ordered with respect to other authentication methods.
- Multiple instances of the SQL authentication module can operate simultaneously, allowing authentication to multiple databases.
- If the database connection drops, it is automatically reestablished after a configurable timeout without Steel-Belted Radius being restarted.
Data from the database can be returned as attributes in the Access-Accept message.

**NOTE:** While Steel-Belted Radius does its best to provide uniformity in the operation of databases from different vendors, differences occur, particularly in the way SQL statements are interpreted. The capabilities of the SQL authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

### SQL Authentication Process

Any RADIUS attribute (or Steel-Belted Radius request variable) from the request can be used in an SQL SELECT statement. Any return list attribute (that is, a Steel-Belted Radius response variable) can be retrieved from a SQL database and returned in a RADIUS access response message.

**Figure 111: SQL Authentication Process**

![SQL Authentication Process Diagram]

### Stored Procedures

A stored procedure is a sequence of SQL statements that form a logical unit and perform a particular task. You can use stored procedures to encapsulate a set of queries or operations that can be executed repeatedly on a database server. For example, you can code operations on an employee database, such as password lookup, as stored procedures that can be executed by application code.

Stored procedures can be compiled and executed with different parameters and results. Stored procedures can use any combination of input parameters (the values passed to the stored procedure at execution time) and output parameters (the values set or returned by the stored procedure to the calling application or environment).
You can write stored procedures for SQL that communicate with Steel-Belted Radius via input and output parameters to implement custom functions. Stored procedures let you use server-side processing on the SQL server to manipulate the information specified by variables. How you use these stored procedures depends on details specific to the implementation of SQL that you are using.

**NOTE:** Do not configure a stored procedure to call the same attribute more than once. Doing so may cause Steel-Belted Radius to fail.

For information on using stored procedures with the Oracle SQL database, see “Working With Stored Procedures in Oracle” on page 202. For information on using stored procedures with the Microsoft SQL database, see “Working With Stored Procedures in MS-SQL” on page 204.

**Connectivity Issues**

Steel-Belted Radius may encounter serious problems if the connection between Oracle and Steel-Belted Radius becomes unstable. The most common reasons for a connection becoming unstable are:

- Slow or unreliable network response times
- Interruptions in connectivity caused by intervening network devices, such as a firewall timing out the connection

To prevent connectivity problems, consider implementation of one of the following solutions:

- To minimize problems caused by intervening firewalls, configure your firewall to pass traffic on the Oracle communications ports between the Steel-Belted Radius server and the Oracle server without restriction.

- To minimize network latency and firewall-related problems, move the Steel-Belted Radius server to the same network segment as the Oracle server.

- If moving your Steel-Belted Radius server is not feasible, locate a second Steel-Belted Radius server on the same network segment as your Oracle server, and configure your current Steel-Belted Radius server to proxy all authentication requests to this new device. This configuration will allow you to open RADIUS ports on the firewall only for the Steel-Belted Radius server (instead of opening RADIUS ports for all network access devices). Because proxy functions in Steel-Belted Radius do not require an uninterrupted connection to process requests, this solution allows you to retain your current firewall timeout settings.
Configuring SQL Authentication

You must configure both Steel-Belted Radius and the SQL database to support SQL authentication. The configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The required transport must be in place between SQL client software and the SQL server.
- The SQL server must be configured via a plug-in to coordinate with SQL client software.
- The Steel-Belted Radius server must be configured to communicate with the SQL client software to interact with the back-end SQL server to perform stored procedures or SQL queries.

**Files**

The following files establish settings for configuring SQL authentication in Steel-Belted Radius. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

**Table 24: SQL Authentication Files**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>radsql.aut</td>
<td>Configures settings for SQL authentication (Solaris).</td>
</tr>
<tr>
<td>sqlauth.aut</td>
<td>Configures settings for SQL authentication (Windows).</td>
</tr>
<tr>
<td>radsqjdbc.aut</td>
<td>Configures settings for SQL authentication using JDBC (Solaris/Linux).</td>
</tr>
</tbody>
</table>

**Using the SQL Authentication Header File**

To configure SQL authentication, you must edit the authentication header files, radsql.aut (Solaris with Oracle), radsqjdbc.aut (Solaris/Linux with JDBC), or sqlauth.aut (Windows/ODBC), which are located in the same directory that contains the Steel-Belted Radius service or daemon. Most of these options may be left at their original settings; however, you must modify certain options to accommodate your own database.

After you complete your changes to the authentication header files and restart Steel-Belted Radius, the InitializationString value that you entered in the [Bootstrap] section of the header file appears in the Authentication Methods tab in the Authentication Policies panel. You can then enable, disable, or prioritize your SQL database like any other authentication method in the list.

**Using Multiple SQL Authentication Methods**

You can configure Steel-Belted Radius to authenticate users against more than one SQL database. Each database that you set up in this way becomes a separate selection in the Authentication Methods tab in the Authentication Policies panel.
To add an additional database, create a new header file with extension .aut in the same directory as radsql.aut (Solaris with Oracle), radsqljdbc.aut (Solaris/Linux with JDBC), or sqlauth.aut (Windows/ODBC). You can give this file any name you like, provided its extension is .aut. At startup, Steel-Belted Radius enumerates all .aut files to create its list of authentication methods.

When creating the new file, start by copying the original .aut file. Be sure to change its InitializationString entry to a unique authentication method name; otherwise, Steel-Belted Radius has no way of distinguishing between the different methods in the authentication methods list.

### Connecting to the SQL Database

Upon startup, the SQL authentication module connects to the database, using settings specified by a connect string specified in the header file. The connect string contains information such as the name and location of the database, and the password required to connect. The connect string is passed to the database client to establish the connection.

While a sample connect string is provided in the original header file, you must configure the Connect entry of the header file with a connect string appropriate to your database.

The password for database access must be provided as part of the connect string. If it is not:

- Solaris/Linux: The connection fails.
- Windows: A pop-up window prompts you to enter the password before making the connection at startup and each time a reconnect is required.

If the initial attempt to connect to the database fails, or if a processing error occurs that the SQL authentication module interprets as a database connection failure, the SQL authentication module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.

The SQL authentication module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

**NOTE:** (Solaris/Linux only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code is displayed in the log. You may need to refer to product-specific documentation to decode this result code. With Oracle on Solaris, you can use the `oerr facility-code error-number` command with a facility code of `ora` from the command shell.
SQL Statement Construction

The authentication transaction is based on an SQL query that returns a password (and possibly other information) based on the name entered by the user attempting to log in.

While a sample SQL query is provided in the original header file, you must configure the SQL entry of the header file with a query appropriate to your database. The query you enter must be either an SQL SELECT or SQL EXECUTE statement that contains additional syntax elements which are preprocessed by the SQL authentication module.

The SQL authentication module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

The SQL statement can be very simple. Basically, all that is required is to look up a password and possibly some optional information based on a username. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; the SQL authentication module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.

Example:

```
SELECT password, profile, fullname FROM usertable WHERE username = %name/63s
```

As shown in this example, a parameter consists of a percent sign (%), the name of the parameter and a format specifier. Table 25 lists SQL statement parameter names.

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning for SQL Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>%OriginalUserName</td>
<td>The original full identification of the user, prior to any processing (that is, user@realm).</td>
</tr>
<tr>
<td>%UserName</td>
<td>The full user identification (user and realm strings) after all stripping and processing has been performed.</td>
</tr>
<tr>
<td>%Name</td>
<td>Synonym for UserName.</td>
</tr>
<tr>
<td>%NASName</td>
<td>The name of the network access device that originated the request. This may be the name of the RADIUS clients entry in the database or the value of the NAS-Identifier or NAS-IP-Address attribute.</td>
</tr>
<tr>
<td>%NASAddress</td>
<td>The address of the network access device, in dotted notation.</td>
</tr>
</tbody>
</table>
Along with these parameters, any RADIUS attribute received in the Access-Request can be referred to by using an at-sign (‘@’) followed by the name of the attribute. If you need to specify a literal at-sign character in an SQL statement, such as in a UserName, you must use two at-signs in a row. For example:

SELECT foo FROM bar WHERE field = 'abc@@xyz'

Likewise, if you need to specify a literal percent character (‘%’) in an SQL statement you must use a two percent characters in a row.

The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash (‘/’), a length, and a type, which for SQL authentication is always ‘s’ for string. For example, if the user’s name is stored in the database as a string of up to 63 bytes, you would enter:

%name/63s

### Password Parameters

Normally, the only parameter you’d include in the SQL statement is %name. The %password parameter is available to support databases containing non-unique usernames. For example, your database might allow two people named “George”; one with password “swordfish”, and the other with password “martha”. You can authenticate them correctly with the following query:

```
SELECT password, profile, fullname FROM usertable
WHERE username = %name and password = %password
```

You must return the password as the first column of the result to perform authentication. If the password is not returned in a password column or as an output parameter, no password authentication is performed.

In the following statement, %name is an input parameter used to look up a record.

```
SELECT profile FROM database WHERE username = %name
```

Since there’s no password output parameter, no password authentication is performed. The [Results] section of the .aut file should look something like the following to work with the above SELECT statement:

<table>
<thead>
<tr>
<th>Item</th>
<th>Meaning for SQL Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>%NASModel</td>
<td>The make/model of the network access device, as specified in the Steel-Belted Radius database.</td>
</tr>
<tr>
<td>%Password</td>
<td>The PAP password.</td>
</tr>
<tr>
<td>%RADIUSClientName</td>
<td>The name of the network access device, as specified in a RADIUS clients entry in the Steel-Belted Radius database.</td>
</tr>
</tbody>
</table>

**NOTE:** Be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a parameter size greater than the actual field size is specified.
[Results]
Password=0
Profile=1/50
Alias=0

If the record cannot be found in the database, the authentication attempt fails.

**NOTE:** If you are not using password checking for authentication, the Password parameter must be set to 0 in the [Results] section.

### Overlapped Execution of SQL Statements

The SQL authentication module is multi-threaded. SQL authentication can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry in the .aut file’s [Settings] section.

If MaxConcurrent is set to 1, SQL execution occurs serially, and the SQL execution for each authentication request must complete before execution for the next request may begin.

By increasing MaxConcurrent, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of MaxConcurrent, possibly even at 1 or 2. The optimum value is a matter of experiment.

**NOTE:** A setting of MaxConcurrent = 1 should be sufficient for all but the most demanding environments. Increase this value only slowly and conservatively.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. So another factor that determines how MaxConcurrent should be set might be the database license.

### %result Parameter

The %result parameter is a string value that can be returned as a column or stored procedure output parameter. The %result parameter can be used with or without password authentication.

The value expected to be returned in this parameter when authenticating a user can be specified in the SuccessResult entry of the [Settings] section. For example, if a user is successfully authenticated by the SQL authentication method, the result signifying success is the text string “okay”. This can be automatically checked by the following setting.

[Settings]
SuccessResult = okay

NOTE: The string comparison is case insensitive.

If the SQL statement succeeds but the SuccessResult value does not match the expected value returned from the database, Steel-Belted Radius issues a reject response, which can include any attributes and values configured in the [FailedSuccessResultAttributes] section of the *.aut file.

If PerformSuccessResultCheckAfterPasswordCheck=1 is specified and the SQL statement performs a password check that fails, Steel-Belted Radius does not process the SuccessResult and does not return the attributes from the [FailedSuccessResultAttributes] section in the reject response. If PerformSuccessResultCheckAfterPasswordCheck=1 is specified and the SQL statement performs a password check that succeeds but the SuccessResult value does not match the value returned from the database, Steel-Belted Radius issues a reject response that contains the attributes from the [FailedSuccessResultAttributes] section.

In the following statement, %password is passed to a stored procedure, which returns a %result of either “okay” or something else (that signifies a rejection):

BEGIN CheckUser(%name, %password, %result); END;

Another example might be a database of usernames, passwords, and account status. The administrator can enable a user by setting the user’s account status to “okay” or disable the user by setting the account status to some other value, without having to delete the record. In the following statement, both password and result columns are checked:

SELECT password, result FROM database WHERE username = %name

[Results]
Password=1/50
%Result=2/50
Profile=0
Alias=0

SQL Authentication and Password Format

Steel-Belted Radius supports the authentication of users residing in a SQL database, in which password values for the users are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1 + Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

Hashed Passwords

Values in the Password column include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces { } and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved from the table Password column, the entire password is assumed to be in clear text format. In summary:

- **PasswordText** indicates clear text format (no encryption)
{crypt}HashHash indicates UNIXcrypt format

{SHA}HashHashHash indicates SHA1 + Base64 hash

{SSHA}HashHashHashSalt indicates salted SHA1 + Base64 hash

{md4}HashHash indicates MD4 hash of the Unicode form of password

{enc-md5}EncryptedEncrypted indicates a reversibly encrypted password

NOTE: Refer to RFC 2759 for details about how MS-CHAP-V2 produces an MD4 hash value.

NOTE: Although Steel-Belted Radius reads passwords encoded in enc-md5 format, you must purchase the Software Developer’s Kit to convert clear-text passwords to this format.

UNIXcrypt is the standard hash algorithm that is used for the /etc/passwd file on Solaris/Linux systems. This may be necessary if, for example, the standard user database on a Solaris or Linux machine (the /etc/passwd file) is migrated to a SQL database, so that the values in the Password column of the SQL table are processed with UNIXcrypt.

Steel-Belted Radius may be configured to expect that the values retrieved from the SQL table Password column during authentication have been run through UNIXcrypt by adding the following entry into the [Settings] section of the SQL authentication header file:

PasswordFormat=3

Automatic Parsing

If PasswordFormat is set to 0, Steel-Belted Radius attempts to determine the password format automatically by parsing it. This is the recommended setting. Automatic parsing expects the password to be stored in one of the formats described in this section.

NOTE: The setting for automatic password parsing in older versions of Steel-Belted Radius (auto) has been deprecated.

Working With Stored Procedures in Oracle

The following notes discuss some considerations specific to Oracle, which uses the term package and package body when referring to stored procedures.

Assume you have a SELECT statement that extracts a user’s name, password, and profile from the table usertable when it receives the user’s name as an input parameter:
SELECT fullname, password, profile FROM usertable WHERE username = %name/63s

To write a package called myPack1 that performs the equivalent function, you would enter the following sequence of commands:

```sql
Package myPack1
  is
  PROCEDURE myProc
  (name  IN  VARCHAR2,
   passOUTVARCHAR2,
   prof OUTVARCHAR2,
   fName OUT VARCHAR2);
End myPack1;
```

When referencing the package from sqlauth.aut, you would point to the package name myPack1 (not the procedure name myProc):

```sql
Package Body myPack1
  is
  PROCEDURE myProc
  (name   IN  VARCHAR2,
   pass   OUT VARCHAR2,
   prof  OUT VARCHAR2,
   fName OUT VARCHAR2)

  IS

  BEGIN

  SELECT fullname INTO fName, password INTO pass, profile INTO prof FROM usertable WHERE username = name;
  END myProc;
End myPack1;
```

When you invoke the stored procedure, you would delineate each parameter as an input (i), output (o), or input/output (io) variable. The presence of an io or o keyword indicates that the value returned from the database is to be included in the Access-Accept response as if it had been coded in the [Results] section. If a r value is included in the suffix (for example, !ir, !r, or !or), the parameter is expected to be an output parameter, and the attribute is to be treated as if it were included in the [FailedSuccessResultAttributes] section. Variables that are not specifically marked are considered input parameters by default.

---

**NOTE:** Do not configure a stored procedure to call the same attribute more than once. Doing so may cause Steel-Belted Radius to fail.

Correct: SQL= {call joeproc2 ( @class!o)}
Incorrect: SQL= {call joeproc2 ( @class1o, @class1o)}

You could replace the SELECT statement by invoking myProc as follows:
When using input-output parameters with Oracle, you must set the DefaultResults setting to 0. Any other variables that need to be returned (such as Reply-Message) must be identified by the “!o” marker within the SQL statement.

---

**Working With Stored Procedures in MS-SQL**

A simple example of a stored procedure returns a result set in the same way as a Select statement. For example, assume you have a table with the following fields: username, password, Alias, and active, where all fields have the datatype varchar. You want a stored procedure that will return a password and alias when the username and password received in the request match entries in the database, provided that active field has a value of yes.

**Example 1**

To create a simple stored procedure, run the following command sequence from MS Query Analyzer to create a stored procedure called `rsp_getpword`.

```sql
CREATE PROCEDURE rsp_getpword
    @Uname varchar(21),
    @pword varchar(21)
AS
    SELECT password, alias FROM authentication WHERE username = @Uname AND password = @pword AND active = 'yes'
GO
```

This stored procedure can then be executed from a *.aut file as follows:

```sql
SQL= Execute rsp_getpword %username, %password
[results]
Password=1
Alias=2
```

**Example 2**

More complex stored procedures take input and output parameters in a manner similar to that used by Oracle. For example, assume you have a table with the following fields: username, password, profile, and active, where all fields have a datatype of varchar. You want a stored procedure that returns a password and profile when the username and password received in the request match a username and password in the database, provided that the active field has a value of yes.

First, to create the stored procedure, run the following command from MS query analyzer:

```sql
CREATE PROCEDURE rsp_authuser
    @uname as varchar(20),
```
@pword as varchar(21) OUTPUT, 
@profile as varchar(21) OUTPUT
AS
SELECT @pword = password, @profile = profile FROM authentication WHERE
username = @uname AND active = 'yes'
GO

This stored procedure can then be executed from a *.aut file as follows:

```
SQL= {call rsp_authuser (%username!, %password!o, %profile!o)}
```

: No entries should be specified in results, everything but the header should be
commented out.
Chapter 16
Configuring SQL Accounting

This chapter presents an overview of SQL accounting and describes how to configure SQL accounting in Steel-Belted Radius.

About SQL Accounting

Steel-Belted Radius can write RADIUS accounting information to an external SQL database, independently of the Steel-Belted Radius accounting log.

To set up an external database for use as a repository for RADIUS accounting data, you must place an .acc database configuration file in the same directory that contains the Steel-Belted Radius service (normally C:\RADIUS\Service) or daemon. This file must be modified to contain specialized information about your enterprise database.

Steel-Belted Radius offers the SQL accounting feature as a plug-in software module. Key features of the SQL plug-in include:

- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- The SQL statement can include a wide variety of arithmetic and string expressions.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- Attribute and other data from the accounting request can be mapped to any parameter of the SQL statement (and hence to any field in the table) by means of a simple syntax.
- Different request types can be mapped to different SQL statements that may operate against distinct tables within the database.
- Multiple instances of a SQL statement can be overlapped for simultaneous execution.
- Multiple instances of the SQL accounting module can operate simultaneously, allowing logging to multiple databases.
If the database connection drops, it is automatically reestablished after a configurable timeout without restarting Steel-Belted Radius.

SQL accounting responses can return information.

Stored procedures invoked by SQL accounting can make use of input parameters, record results, and return output parameters.

**NOTE:** While Steel-Belted Radius tries to provide uniformity in the operation of databases from different vendors, differences exist, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

### Stored Procedures

A stored procedure is a sequence of SQL statements that form a logical unit and perform a particular task. You can use stored procedures to encapsulate a set of queries or operations that can be executed repeatedly on a database server. For example, you can code operations on an employee database, such as password lookup, as stored procedures that can be executed by application code.

Stored procedures can be compiled and executed with different parameters and results. Stored procedures can use any combination of input parameters (the values passed to the stored procedure at execution time) and output parameters (the values set or returned by the stored procedure to the calling application or environment).

You can write stored procedures for SQL that communicate with Steel-Belted Radius via input and output parameters to implement custom functions. Stored procedures let you use server-side processing on the SQL server to manipulate the information specified by variables. How you use these stored procedures depends on details specific to the implementation of SQL that you are using.

For information on using stored procedures with the Oracle SQL database, see “Working With Stored Procedures in Oracle” on page 202. For information on using stored procedures with the Microsoft SQL database, see “Working With Stored Procedures in MS-SQL” on page 204.

### Connectivity Issues

Steel-Belted Radius may encounter serious problems if the connection between Oracle and Steel-Belted Radius becomes unstable. The most common reasons for a connection becoming unstable are:

- Slow or unreliable network response times
- Interruptions in connectivity caused by intervening network devices, such as a firewall timing out the connection

To prevent connectivity problems, consider implementation of one of the following solutions:
To minimize problems caused by intervening firewalls, configure your firewall to pass traffic on the Oracle communications ports between the Steel-Belted Radius server and the Oracle server without restriction.

To minimize network latency and firewall-related problems, move the Steel-Belted Radius server to the same network segment as the Oracle server.

If moving your Steel-Belted Radius server is not feasible, locate a second Steel-Belted Radius server on the same network segment as your Oracle server, and configure your current Steel-Belted Radius server to proxy all authentication requests to this new device. This configuration will allow you to open RADIUS ports on the firewall only for the Steel-Belted Radius server (instead of opening RADIUS ports for all network access devices). Because proxy functions in Steel-Belted Radius do not require an uninterrupted connection to process requests, this solution allows you to retain your current firewall timeout settings.

Configuring SQL Accounting

You must configure both Steel-Belted Radius and the SQL database to support SQL accounting. The configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The SQL server must be configured to be listening for client requests. Note that for SQL purposes, the Steel-Belted Radius server must be a client of the SQL server.
- The Steel-Belted Radius server must know the machine where the SQL server software runs, and it must know the protocol and port used in communicating with that machine.
- The required transport must be in place between SQL client and server.

Files

The following files establish settings for configuring SQL accounting in Steel-Belted Radius. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>radsql.acc</td>
<td>Configures settings for SQL accounting (Solaris and Oracle).</td>
</tr>
<tr>
<td>radsqljdbc.acc</td>
<td>Configures settings for SQL accounting (Solaris/Linux and JDBC).</td>
</tr>
<tr>
<td>sqlacct.acc</td>
<td>Configures settings for SQL accounting (Windows/ODBC).</td>
</tr>
</tbody>
</table>
Using the SQL Accounting Header File

To configure SQL accounting, you must edit the accounting header file (radsql.acc (Solaris), radsqljdbc.acc (Solaris/Linux and JDBC) or sqlacct.acc (Windows/ODBC)), located in the same directory that contains the Steel-Belted Radius service (normally C:\RADIUS\Service) or daemon.

You must modify certain options in the accounting header file to accommodate your own database. After you update your accounting header file and restart Steel-Belted Radius, accounting proceeds as you have configured it.

Using Multiple SQL Databases

You can configure Steel-Belted Radius to log accounting transactions against more than one SQL database.

To add an additional database, create a new header file with extension .acc in the same directory as radsql.acc (Solaris), radsqljdbc.acc (Solaris/Linux and JDBC), or sqlacct.acc (Windows/ODBC). You can give this file any name you like, provided its extension is .acc. At startup, Steel-Belted Radius enumerates all .acc files to create its list of accounting modules.

NOTE: When creating the new file, start by duplicating the original .acc file, then make whatever modifications are necessary.

Connecting to the SQL Database

Upon startup, the SQL accounting module connects to the database, based on a connect string specified in your accounting header file. The connect string contains information such as the name and location of the database, and the password required to connect. The connect string is passed to the database client to establish the connection.

While a sample connect string is provided in the original header file, you must configure the Connect entry of the header file with a connect string appropriate to your database.

The password for database access must be provided as part of the connect string or the following results occur:

- Solaris/Linux: The connection fails.
- Windows: A pop-up window prompts the user to enter a password before making the connection at startup and each time a reconnect is required.

If the initial attempt to connect to the database fails, or if a processing error occurs that the SQL accounting module interprets as a database connection failure, the SQL accounting module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.
The SQL accounting module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

**NOTE:** (Solaris/Linux only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code appears in the log. You may need to refer to product-specific documentation to decode this result code. With Oracle on Solaris, you can use the `oerr facility-code error-number` command with a facility code of ora from the command shell.

---

**SQL Statement Construction**

For each accounting request whose Acct-Status-Type is mapped to a SQL statement, that accounting request is logged to the backend database by executing the associated SQL statement.

While a sample SQL statement is provided in the original header file, you must configure one or more SQL entries of the header file with a statement appropriate to your database. Each SQL statement is typically an `INSERT INTO` statement and may contain additional syntax elements that are preprocessed by the SQL accounting module.

The SQL accounting module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

A SQL statement can be very simple. Basically, all that is required is to set fields of the database record with values from the request. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; The SQL accounting module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.
**INSERT Statement and VALUES Section**

The following is an example of a SQL INSERT statement that might be found in a Steel-Belted Radius .acc file.

**Figure 112: INSERT Statement Example**

```
INSERT INTO usagelog (Time, NASAddress, SessionID, Type, Name, BytesIn, BytesOut) VALUES (%TransactionTime/t, %NASAddress, @Acct-Session-Id, @Acct-Status-Type, %FullName/40s, @Acct-Input-Octets, @Acct-Output-Octets)
```

In the VALUES section, the names (between parentheses) represent the values inserted into the SQL table columns. To support the SQL accounting module, each item in the VALUES section must be prefixed with a @ sign or a % sign.

- @ indicates a RADIUS accounting attribute. The attribute name must also be listed in the account.ini file. This remains true even if the account.ini file is disabled.

- % indicates an item associated with the INSERT request that is not a RADIUS accounting attribute. Table 27 lists the Steel-Belted Radius items that may be provided.

**Table 27: Insert Statement Syntax**

<table>
<thead>
<tr>
<th>Item</th>
<th>Data Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>%TransactionTime</td>
<td>Time</td>
<td>The date/time that the event occurred that is the subject of the request. <strong>NOTE:</strong> You should include the /t (timestamp) data type qualifier with the %TransactionTime argument in SQL statements. If you do not, the %TransactionTime output is formatted as character, with differing results on JDBC and Oracle.</td>
</tr>
<tr>
<td>%Time</td>
<td>Time</td>
<td>The date/time when the request is being processed. (This is later than %TransactionTime if the request is a retry.) <strong>NOTE:</strong> You should include the /t (timestamp) data type qualifier with the %time argument in SQL statements. If you do not, the %Time output is formatted as character, with differing results on JDBC and Oracle.</td>
</tr>
<tr>
<td>%Type</td>
<td>String</td>
<td>The RADIUS accounting request type.</td>
</tr>
<tr>
<td>%NASAddress</td>
<td>IP address</td>
<td>The IP address of the requesting RAS.</td>
</tr>
<tr>
<td>%NASName</td>
<td>String</td>
<td>The name of the network access device that originated the request. This may be the name of the RADIUS client entry in the database or the value of the NAS-Identifier or NAS-IP-Address attribute.</td>
</tr>
<tr>
<td>%NASModel</td>
<td>String</td>
<td>The RAS make/model.</td>
</tr>
<tr>
<td>%FullName</td>
<td>String</td>
<td>The full name of the logged in user.</td>
</tr>
<tr>
<td>%AuthType</td>
<td>String</td>
<td>The method by which the user was authenticated.</td>
</tr>
<tr>
<td>%RADIUSClientName</td>
<td>String</td>
<td>The name of the network access device, as specified in a RADIUS client entry in the Steel-Belted Radius database.</td>
</tr>
</tbody>
</table>
A format specifier may appear immediately following each parameter. The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash (’/’), possibly a length, and a data type. Table 28 lists the available data types.

**Table 28: Data Types**

<table>
<thead>
<tr>
<th>Format Specifier</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>/xs</td>
<td>A text string of length x. /s indicates a string with the default length of 256.</td>
</tr>
<tr>
<td>/xb</td>
<td>A binary data string of length x. A binary string is different from an text string in that it is not NULL-terminated and is not restricted to ASCII characters. /b indicates a binary data string with the default length of 256.</td>
</tr>
<tr>
<td>/n</td>
<td>32-bit integer</td>
</tr>
<tr>
<td>/n8</td>
<td>8-bit integer</td>
</tr>
<tr>
<td>/n16</td>
<td>16-bit integer</td>
</tr>
<tr>
<td>/n32</td>
<td>32-bit integer (same as /n)</td>
</tr>
<tr>
<td>/nxx</td>
<td>Integer xx bits in length. For example, /n64 indicates a number with a length of 64 bits.</td>
</tr>
<tr>
<td>/t</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>

**NOTE:** Steel-Belted Radius supports integers larger than 32 bits by manipulating them as binary data strings. The Solaris Oracle 8 plug-ins are able to convert binary data strings between Oracle VARRAW types (/xb) and Oracle NUMBER types (/n). Oracle types must be declared with enough precision to avoid truncation when inserting into the database, and care must also be taken to avoid truncation when retrieving from the database. In particular, avoid retrieving Oracle VARRAW types larger than 256 bytes. Other database/operating-system combinations may not allow for integers larger than 32 bits.

If a format specifier is not present in the SQL statement syntax, Steel-Belted Radius automatically defaults to an appropriate specifier based on the actual parameter type. For example, @Acct-Input-Octets is a number, and defaults to /n.

**NOTE:** For strings, always include a format specifier, and be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a length greater than the actual field size is specified. If no format specifier is present, the length defaults to 256 characters, which may cause the compilation to fail.

Steel-Belted Radius automatically attempts to convert between the internal format of a parameter and its format in the database, as described by the format specifier. In most cases, the formats are equivalent; if not, Steel-Belted Radius performs reasonable conversions.
Table 29 lists the internal formats and their compatible database formats:

<table>
<thead>
<tr>
<th>Internal Format</th>
<th>Compatible Database Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary data string</td>
<td>/b, /xb, /ln, /ln8, /ln16, /ln32</td>
</tr>
<tr>
<td>Number</td>
<td>/n, /ln8, /ln16, /ln32, /lx, /ls</td>
</tr>
<tr>
<td>String</td>
<td>/xs, /ls</td>
</tr>
<tr>
<td>Time (seconds since 1/1/70)</td>
<td>/t, /ln, /ln32, /lx, /ls</td>
</tr>
<tr>
<td>IP address</td>
<td>/ln, /ln32, /lx, /ls</td>
</tr>
</tbody>
</table>

As you write the INSERT statement for your SQL accounting header file (.acc), we recommend the following syntax checklist:

- The column names and their corresponding attributes in the VALUES section are order-dependent. In the example shown in Figure 112 on page 212, the %TransactionTime/t value would be inserted into the Time column (and formatted as a timestamp), the %NASAddress value would be inserted into the NASAddress column, and so forth. The ordering of these settings is critical to proper RADIUS accounting data insertion, since each column in the SQL table may be a specific data type, such as varchar or int.

- The use of left and right parentheses ⟨( ⟩, the backslash ⟨\⟩, the forward slash ⟨/⟩ and even blank spaces are all extremely important and must be exact. You can add as many columns and attributes as you want for your RADIUS accounting needs; however, be sure to model your INSERT statement syntax on the example shown in Figure 112 on page 212.

- An attribute listed incorrectly in the VALUES section, such as @Acct_Session-Id rather than @Acct-Session-Id, causes the SQL statement to fail during a RADIUS accounting transaction. The attribute’s syntax must match its corresponding attribute name in the account.ini file, which in turn matches the attribute’s name in the appropriate dictionary file, which allows Steel-Belted Radius to process the attribute correctly when it is received from the RAS (the RADIUS client).

- An attribute listed in the VALUES section that is missing its prefix of ‘@’ or ‘%’ causes the SQL statement to fail during a RADIUS accounting transaction.

- If a carriage return is present within the INSERT statement without the backslash ⟨\⟩ to indicate the end of the line, the SQL statement fails during a RADIUS accounting transaction.

- Do not make the lines in the .acc file too long. There is a line length limit of 255 characters. Use the backslash ⟨\⟩ to indicate the end of the line before that limit is reached. If a line exceeds this limit, the SQL statement fails during a RADIUS accounting transaction.
Using Multiple SQL Statements

The most common use of accounting is to track user sessions. However, accounting requests are generated when the RAS starts up and shuts down; and, vendor-specific uses of accounting are used to track other RAS phenomena. Clearly, it might be advisable to log different types of accounting events to different tables.

The Acct-Status-Type attribute of an accounting request indicates the request type. You may, if you like, create multiple SQL statements, and map each Acct-Status-Type to one of these SQL statements. The different statements may update different tables in the database, but they all share the single database connection.

Overlapped Execution of SQL Statements

The SQL accounting module is multi-threaded. SQL accounting can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry in the .acc file’s [Settings] section.

If MaxConcurrent is set to 1, SQL execution occurs serially, and the SQL execution for each accounting request must complete before execution for the next request may begin.

By increasing MaxConcurrent, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of MaxConcurrent, possibly even at 1 or 2. You can find the optimum value for your system by experimentation.

NOTE: A setting of MaxConcurrent = 1 should be sufficient for all but the most demanding environments. Increase this value slowly and conservatively.

MaxConcurrent determines the maximum overlap for executing any single SQL statement. Multiple SQL statements for different request types are not interdependent, and executions of one statement do not affect executions of a different statement.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. The database license may also have an influence on the optimum setting for MaxConcurrent.

SQL Accounting Return Values

SQL accounting statements can return information in RADIUS attributes in an accounting response. This is useful only if you are using a client that expects and supports attributes embedded in a RADIUS accounting response message.

Stored procedures can also return output parameters. The way in which these stored procedures are called depends on your operating system:
To call an Oracle stored procedure in a Solaris environment:

\[
\text{BEGIN storedProcedure(parameters...); END;}
\]

To call an Oracle stored procedure in a Windows environment:

\[
call(storedProcedure(parameters...))
\]

---

**Accounting Stored Procedure Example**

A simple stored procedure can return a result set in the same way as a `Select` statement. For example, assume you have a table with the following fields: `username`, `password`, `Alias`, and `active`, where all fields have the datatype `varchar`. You want a stored procedure that will return a password and alias when the username and password received in the request match entries in the database, provided that `active` field has a value of yes'.

The following example executes a stored procedure to update an accounting table in Steel-Belted Radius.

1. Create an accounting table by executing the following command:

   ```
   create table accounting
   ( TransactionDate varchar(20), Username varchar(21), SessionID varchar(12),
   NASIPAddr varchar(15), NASPort varchar(5), UserIPAddr varchar(15), CallingNum varchar(12),
   CalledNum varchar(12),
   type varchar(4), SessionTime varchar(14), Disconnect varchar(12) )
   ```

2. Create a `rsp_account` stored procedure that can be called by a `*.acc` file.

   ```
   create procedure rsp_account
   @transactiontime varchar(21),
   @username varchar(21),
   @AcctSessionID varchar(21),
   @NASIPAddress varchar(21),
   @NASPORTTYPE varchar(21),
   @FRAMEDIPADDRESS varchar(21),
   @callingstationid varchar(21),
   @calledstationid varchar(21),
   @TYPE varchar(21),
   @ACCTSESSIONTIME varchar(21),
   @ACCTTERMINATIONCAUSE varchar(21)
   AS
   INSERT INTO Accounting (TransactionDate, username, SessionID, NASIPAddr, NASPort, UserIPAddr, CallingNum, CalledNum, type, SessionTime, Disconnect)
   VALUES (@transactiontime, @username, @AcctSessionID, @NASIPAddress, @NASPORTTYPE, @FRAMEDIPADDRESS, @callingstationid, @calledstationid, @TYPE, @ACCTSESSIONTIME, @ACCTTERMINATIONCAUSE)
   ```

3. Create the `mysqlacct.acc` file to call the `rsp_account` stored procedure.
Chapter 16: Configuring SQL Accounting

Note that the mysqlacct.acc file uses an SQL=EXECUTE `procedure_name` `value1,...,valueN` statement instead of an SQL=INSERT into `table` `(column1, ...` `columnN)` `Values (value1,...,valueN)`, since the stored procedure will do the INSERT action. You would configure the CONNECT statement to reflect your operating environment.

[Bootstrap]
 LibName=sqlacct.dll
 Enable=1
 InitializationString=

[Settings]
 Connect=DSN=<dsn_name_here>;UID=<username_for_dB>;PWD=<password_for_dB>
 ConnectTimeout=25
 WaitReconnect=2
 MaxWaitReconnect=360
 ParameterMarker=?
 loglevel=2

[Type]
 1=User
 2=User
 3=User

[Type/User]
 SQL=Execute rsp_account  %transactiontime/t, \ 
   @user-name/21s, \ 
   @Acct-Session-ID/12s, \ 
   @NAS-IP-Address/15s, \ 
   @NAS-PORT-TYPE/5s, \ 
   @FRAMED-IP-ADDRESS/15s, \ 
   @calling-station-id/12s, \ 
   @called-station-id/12s, \ 
   %TYPE/4s, \ 
   @ACCT-SESSION-TIME/14s, \ 
   @ACCT-TERMINATION-CAUSE/12s
 ConcurrentTimeout=30
 MaxConcurrent=2
Chapter 17

Configuring LDAP Authentication

This chapter presents an overview of LDAP authentication and describes how to configure LDAP authentication in Steel-Belted Radius.

About LDAP Authentication

Steel-Belted Radius can authenticate against records stored in an external LDAP database. Any attribute(s), such as username and password, can be used to query the database.

External database authentication is typically used when an organization has a large amount of user information stored in an LDAP database, and wants to authenticate these users using RADIUS. Authentication against an existing LDAP database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database.

Steel-Belted Radius offers LDAP authentication as a plug-in software module. Key features of the LDAP plug-in include the following:

- LDAP Version 3 is supported.
- SSL is supported if you have Netscape certificates.
- You can authenticate via LDAP Bind or via a password returned from an LDAP Search request (BindName).
- A single Search request or a sequence of Search requests can be specified.
- Bind, Base, and Search strings can include variables.
- New Bind parameters can be specified during a sequence of searches.
- Other authentication credentials can be specified in a string that can include variable values.
- Variables may be set from the RADIUS request packet and from LDAP Search results.
- Variables may be used to specify RADIUS response attributes and other response information.
The RADIUS response can include RADIUS attributes found in the LDAP database, or it can reference a Steel-Belted Radius profile or user entry.

Several features similar to SQL authentication are supported, such as round-robin load balancing, the “server of last resort,” and activation targets.

The variable table allows both attributes and %Profile in the [Response] section.

**LDAP Variable Table**

The LDAP Variable Table lets you translate a RADIUS request into an LDAP lookup. At the beginning of each LDAP authentication request, Steel-Belted Radius creates a Variable Table. Attributes and other information from the RADIUS request are entered in the Variable Table for use in LDAP Bind, Base, and Search strings. When attributes are returned by LDAP requests, they too are entered in the Variable Table. Finally, selected information from the Variable Table is returned to the RADIUS client in the RADIUS response packet.

**Figure 113: Role of the Variable Table in LDAP Authentication**

**Types of LDAP Authentication**

To design an LDAP authentication method, consider how you want to validate the username and password.
The LDAP plug-in offers two techniques for validating the username and password. Each header file that you write to control LDAP authentication must employ **Bind** or **BindName**. The differences between the two techniques have to do with how Steel-Belted Radius connects to the LDAP server and whether the username/password validation is performed by the LDAP server or by Steel-Belted Radius.

### BindName Authentication

When you use **BindName** authentication, your LDAP header file provides Steel-Belted Radius with the username and password of an account on the LDAP server. This must be an account that has privileges to access all of the information that you require to authenticate users. In the LDAP header file, you provide the username in the **BindName** parameter, and the password in the **BindPassword** parameter.

After you complete the LDAP header file, each time Steel-Belted Radius starts up, it executes a **Bind** request to the LDAP server using the **BindName** and **BindPassword** parameters as its credentials. If the LDAP server can validate these credentials, a connection is established between the two servers. This connection remains “up” all the time. It is disconnected only if the Steel-Belted Radius server or the LDAP server goes down, and it’s re-established as soon as possible after the “down” server comes back up. The LDAP header file offers a number of connection and re-connection timeouts and other parameters that regulate this relationship.

Any time authentication via LDAP is required, Steel-Belted Radius consults the corresponding LDAP header file. When you use **BindName** authentication, this file must contain a Search command that maps the username from the Access-Request to a password attribute in the LDAP database. The Search may retrieve other LDAP attributes as well. When the Search returns its results, Steel-Belted Radius compares the value of the password returned from the LDAP database with the password from the incoming Access-Request. If the two values are the same, the password is considered validated.

When the connection to the LDAP server is established using **BindName**, multiple authentications can be performed at the same time over the same connection.

### Bind Authentication

When you use **Bind** authentication, Steel-Belted Radius authenticates connection requests by attempting to **Bind** to the LDAP server using the username and password from the incoming Access-Request or from a configured username and password. If this **Bind** request succeeds, the password is validated. This is essentially “pass-through” authentication; Steel-Belted Radius presents an LDAP user’s credentials to the LDAP server and asks to have them validated.

In the simplest case, a single connection is established for each Access-Request and is kept open only long enough for the LDAP server to validate the password and respond to any Search requests. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.
A more sophisticated search technique can take advantage of flexible **Bind**, which allows you to allocate a sequence of connections for each Access-Request. Each in turn is kept open only long enough for the server to process each search criterion. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.

**Attributes and LDAP Authentication**
A username and password may be all the information that you require to authenticate users. However, the LDAP plug-in offers a number of techniques for working with checklist and/or return list attributes, should you need them.

---

**Configuring LDAP Authentication**

To configure an LDAP authentication method, you must edit the header file that controls the LDAP authentication sequence.

Table 30 summarizes the process of configuring an LDAP authentication method for Steel-Belted Radius. It lists the sections that you must edit in the header file to accomplish each step. No step may be omitted. You must at least consider the entries that you want to put in each section of the header file, even if you decide to leave most of that section blank.

**Table 30: LDAP Authentication Header File Topics**

<table>
<thead>
<tr>
<th>Step</th>
<th>LDAP Configuration Task</th>
<th>.aut File Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Decide how you want Steel-Belted Radius to validate RADIUS access requests. Two major areas of choice are described above: (1) <strong>Bind</strong> or <strong>BindName</strong>; and (2) Profile, Alias, or attribute list.</td>
<td>All sections</td>
</tr>
<tr>
<td>2</td>
<td>Determine which incoming RADIUS attributes are required to perform the LDAP search.</td>
<td>[Response]</td>
</tr>
<tr>
<td>3</td>
<td>Determine which LDAP attributes support are required to perform the LDAP search.</td>
<td>[Attribute/<em>name</em>]</td>
</tr>
<tr>
<td>4</td>
<td>Design Search template(s) that can find the necessary data in your LDAP database schema.</td>
<td>[Search/<em>name</em>]</td>
</tr>
<tr>
<td>5</td>
<td>Extract the data from the incoming RADIUS packet that Steel-Belted Radius will use to perform the LDAP <strong>Bind</strong> and <strong>Search</strong> requests.</td>
<td>[Request]</td>
</tr>
<tr>
<td>6</td>
<td>Select defaults that you want Steel-Belted Radius to use when corresponding values are not provided.</td>
<td>[Defaults]</td>
</tr>
<tr>
<td>7</td>
<td>Enable connections between the Steel-Belted Radius server and LDAP server(s).</td>
<td>[Server] [Server/<em>name</em>] [Settings] [Failure]</td>
</tr>
<tr>
<td>8</td>
<td>Enable the LDAP plug-in and name the authentication method.</td>
<td>[Bootstrap]</td>
</tr>
</tbody>
</table>

The order in which you should edit header file sections is the reverse order in which Steel-Belted Radius processes them. The processing sequence is described in “LDAP Authentication Sequence” on page 226.
Supporting Secure Sockets Layer
You must follow the instructions below for SSL to be supported by the LDAP plug-in:

1. Set SSL in the [Settings] (or [Server/name]) section to 1.

2. Set the Certificates field in the [Settings] (or [Server/name]) section to the path where the cert7.db and key3.db files are located.
   
   The name of the files (that is cert7.db or key3.db) should not be included.

3. Set the port in the [Server] section to the SSL port of the LDAP server.

Files

The following file establishes settings for LDAP authentication. For more information about this file, refer to the Steel-Belted Radius Reference Guide.

<table>
<thead>
<tr>
<th>Table 31: LDAP Authentication Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>ldapauth.aut</td>
</tr>
</tbody>
</table>

LDAP Database Schema

The most important factor in the success of your LDAP authentication methods is the design of your LDAP database schema. It’s assumed that you already have a schema in place.

Often, you can use the LDAP plug-in without changing the LDAP database schema at all. In Figure 114, the user record already provides an LDAP attribute called Organization. If you intend to grant connection privileges according to the organization to which each user belongs, you can create profiles in the Steel-Belted Radius database whose names match the strings you are already using for the Organization attribute. You can then create an LDAP authentication header file that retrieves the value of the Organization attribute from the LDAP database and returns it to Steel-Belted Radius as the name of the profile to use.

**NOTE:** If you are using BindName authentication, you need to be able to identify which LDAP attribute contains the user’s password. In the schema below, this attribute is called User-Secret.
When the authentication strategy you have chosen requires data that is not currently in the schema, you might need to modify the schema.

The name of a Steel-Belted Radius profile is a typical example. Consider the example shown in Figure 114. If you want to assign connection privileges to users in some way other than by Organization, and no other LDAP attribute seems appropriate, you can add an LDAP attribute that names a profile. In Figure 115, this attribute is called RADIUS-Profile. This attribute contains a string value that can be set to the name of a profile defined in the Steel-Belted Radius database.
Chapter 17: Configuring LDAP Authentication

Figure 115: Modifying a Schema to Enhance LDAP Authentication

### LDAP Authentication and Password Format

Steel-Belted Radius supports authentication of users whose records reside in an LDAP table in which password values are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1 + Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

### Hashed Passwords

Encoded values include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces ‘{" ‘}’ and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved, the entire password is assumed to be in clear text format. In summary:

- **PasswordText** indicates clear text format (no encryption)
### LDAP Authentication Sequence

The sequence of an LDAP authentication transaction is controlled by the LDAP authentication header file as follows:

1. The Variable Table is initialized to default values as specified in the [Defaults] section. All variables that are not listed in the [Defaults] section are initialized to null values.

2. The values of RADIUS attributes in the Access-Request are copied to the Variable Table, as specified in the [Request] section.

---

**NOTE:** The setting for automatic password parsing in previous versions of Steel-Belted Radius (auto) has been deprecated.
3. If a Bind entry was specified in the [Settings] section, authentication via LDAP Bind is now performed. The Bind entry is used as a template to construct a bind string, using replacement values from the Variable Table. An LDAP Bind request is then performed to authenticate the user.

4. An LDAP Search request is performed for each [Search/name] section specified. You may specify zero or more separate Search requests.

For each Search request, LDAP Base and Filter strings are constructed from templates, using replacement values from the Variable Table. These Base and Filter strings are then transmitted to the LDAP server in a Search request.

Each attribute/value pair returned by the LDAP Search is used to set the value of the corresponding entry in the Variable Table. Also, the DN returned by the search may be used to set a variable.

5. If a %Password entry appears in the [Response] section, authentication is now performed. The password entered by the user is validated against the value that appears in the %Password variable, and the user is rejected if the passwords don’t match.

6. If a %Profile entry appears in the [Response] section, the value of the %Profile variable is used to look up a Profile entry in the Steel-Belted Radius database. The checklist and return list attributes in that Profile are used to validate the request and return an appropriate response.

7. If a %Alias entry appears in the [Response] section, the value of the %Alias variable is used to look up a Native User entry in the Steel-Belted Radius database. The current transaction is treated as if it came from the “alias” user; that is, the checklist and return list attributes of the alias user are used to validate the request and return an appropriate response.

8. If neither a %Profile nor a %Alias entry appears in the [Response] section, then RADIUS attributes for the response packet are created from the Variable Table, based on attribute entries in the [Response] section.

LDAP Authentication Examples

This topic provides examples of LDAP authentication header file syntax. The examples illustrate how you might:

- Authenticate passwords (Bind or BindName).
- Specify checklist and return list attributes (list the attributes or name a profile entry in the Steel-Belted Radius database).

Bind Authentication with Default Profile

The following example is a simple LDAP authentication header file. Every user is authenticated using a Bind request to the LDAP database. The same Steel-Belted Radius attribute profile is applied to every Access-Request.

[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
Bind=uid=<User-Name>, ou=Special Users, o=bigco.com
LogLevel = 2
UpperCaseName = 0
PasswordCase=original
SSL = 0

[Server]
s1=

[Server/s1]
Host=199.185.162.147
Port = 389

[Defaults]
TheUserProfile = Sample

[Request]
%User-Name = User-Name

[Response]
%Profile = TheUserProfile

[Search/DoLdapSearch]
Base = ou=Special Users, o=bigco.com
Scope = 2
Filter = uid=<dialup>
Attributes = AttrList
Timeout = 20
%DN = dn

[Attributes/AttrList]

If the [Response] section was empty, Steel-Belted Radius would pass the Bind results (accept or reject) directly to its client and no additional RADIUS attributes would be returned in the Access-Response.

**BindName Authentication with Callback Number Returned**

In the following example, requests are authenticated using Search. **BindName** and **BindPassword** values are supplied to permit a connection to the LDAP database. Return list attributes for authentication are listed in the [Response] section. In this example, the network access device needs a callback number to complete the connection. The value of the incoming DNIS attribute Calling-Station-ID is used to ensure that the callback number is the number from which the user's request originated.

---

**NOTE:** This example is incomplete; it omits the [Bootstrap] and [Settings] sections to save space.
Chapter 17: Configuring LDAP Authentication

[Server]
s1=

[Server/s1]
Host = 67.186.4.3
Port = 389
BindName=uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot
BindPassword=ourlittlesecret
Search = DoLdapSearch

[Defaults]
SendThis = DidLDAPAuthSearch

[Request]
%UserName = dialup
Calling-Station-ID = thenumbertocall

[Search/DoLdapSearch]
Base = ou=Special Users, o=bigco.com
Scope = 2
Filter = uid=<dialup>
Attributes = AttrList
Timeout = 20
%DN = dn

[Attributes/AttrList]
dialuppassword

[Response]
%Password = dialuppassword
Reply-Message = SendThis
Ascend-Callback-No = thenumbertocall

LDAP Bind with Profile Based on Network Access Device

In the following example, requests are authenticated using Bind. Checklist and return list attributes for authentication are provided by referencing a profile entry in the Steel-Belted Radius database. The profile to be used depends on the specific network access device from which the user's request originates. Steel-Belted Radius retrieves the profile name by the LDAP database for an IP address that matches the address of the requesting RAS. If this search fails, a profile called limited is used. If a profile name is successfully retrieved from the LDAP database, but no profile by that name can be found in the Steel-Belted Radius database, authentication fails due to “lack of resources” and the user is rejected.

NOTE: This example is incomplete; it omits the [Bootstrap] section and many [Settings] entries to save space.

[Settings]
Bind=uid=<loginID>, ou=Special Users, o=bigco.com
Search = DoLdapSearch

[Server]
s1=
[Server/s1]
Host = 67.186.4.3
Port = 389

[Request]
%UserName = loginID
%NASAddress = deviceIP

[Defaults]
%Profile = limited

[Search/DoLdapSearch]
Base = ou=CommServers, o=bigco.com
Scope = 1
Filter = ipaddr=<deviceIP>
Attributes = AttrList
Timeout = 20
%DN = dn

[Attributes/AttrList]
profile

[Response]
%Profile = profile
Chapter 18
Displaying Statistics

The Statistics panel lets you display summary statistics for authentication, accounting, and proxy forwarding transactions. You can also use the Statistics panel to see how long Steel-Belted Radius has been running and to display a list of the users currently connected through a RAS or tunnel.

Displaying Authentication Statistics

Authentication statistics (Figure 116) summarize the number of authentication acceptances and rejections, with summary totals for each type of rejection or retry.

To display authentication statistics, open the Statistics panel, click the System tab, pull down the View list, and choose Authentication.

Figure 116: Statistics Panel: Authentication Statistics

Table 32 explains the authentication statistics fields and describes possible causes for authentication rejections.
### Displaying Accounting Statistics

Accounting statistics provide information such as the number of transaction starts and stops and the reasons for rejecting attempted transactions. The start and stop numbers rarely match, as many transactions can be in progress at any given time.

<table>
<thead>
<tr>
<th>Authentication Statistic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions Accepts</td>
<td>The current, average, and peak number of RADIUS transactions that resulted in an accept response.</td>
</tr>
<tr>
<td>Rejects</td>
<td>The current, average, and peak number of RADIUS transactions that resulted in a reject response. These are detailed in the Reject Details section.</td>
</tr>
<tr>
<td>Silent Discards</td>
<td>The number of requests in which the client could not be identified. This might occur if a RADIUS client entry cannot be found for a device with the name and/or IP address of a device requesting authentication services.</td>
</tr>
<tr>
<td>Total Transactions</td>
<td>The sum of the accept, reject, and silent discard totals.</td>
</tr>
<tr>
<td>Reject Details Dropped Packet</td>
<td>The number of RADIUS authentication packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.</td>
</tr>
<tr>
<td>Invalid Request</td>
<td>The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.</td>
</tr>
<tr>
<td>Failed Authentication</td>
<td>The number of failed authentication requests, where the failure is due to invalid username or password. If all transactions are failing authentication, the problem might be that the shared secret entered into Steel-Belted Radius does not match the shared secret entered on the client device.</td>
</tr>
<tr>
<td>Failed on Checklist</td>
<td>The number of requests that were authenticated but failed to meet the checklist requirements.</td>
</tr>
<tr>
<td>Insufficient Resources</td>
<td>The number of rejects due to a server resource problem.</td>
</tr>
<tr>
<td>Proxy Failure</td>
<td>The number of rejects that had to be issued because Proxy forwarding to another RADIUS server failed.</td>
</tr>
<tr>
<td>Rejected by Proxy</td>
<td>The number of rejects due to receiving a reject response from a proxy RADIUS target server.</td>
</tr>
<tr>
<td>Retries Received</td>
<td>The number of transactions for which one or more duplicates was received.</td>
</tr>
<tr>
<td>Total Retry Packets</td>
<td>The number of duplicate packets received.</td>
</tr>
<tr>
<td>Challenges</td>
<td>The number of challenges received.</td>
</tr>
</tbody>
</table>
To display authentication statistics, open the Statistics panel, click the System tab, pull down the View list, and choose Accounting.

**Figure 117: Statistics Panel: Accounting Statistics**

Figure 33 describes the accounting statistics and describes possible causes for accounting errors.

**Table 33: Accounting Statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions</td>
<td>The current, average, and peak number of transactions in which a dial-in connection was started following a successful authentication.</td>
</tr>
<tr>
<td>Starts</td>
<td>The current, average, and peak number of transactions in which a dial-in connection was terminated.</td>
</tr>
<tr>
<td>Stops</td>
<td>The number of Accounting-On messages received, indicating that a RADIUS client has restarted.</td>
</tr>
<tr>
<td>Ons</td>
<td>The number of Accounting-Off messages received, indicating that a RADIUS client has shut down.</td>
</tr>
<tr>
<td>Offs</td>
<td>The sum of the start, stop, on and off totals.</td>
</tr>
<tr>
<td>Total</td>
<td>The number of RADIUS accounting packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.</td>
</tr>
<tr>
<td>Dropped Packet</td>
<td>The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.</td>
</tr>
</tbody>
</table>
Proxied request statistics provide information such as the number of proxy authentication or accounting requests and the reasons for any transaction failures that occur.

To display proxied request statistics, open the Statistics panel, click the System tab, pull down the View list, and choose Proxied Requests.

**Table 33: Accounting Statistics (continued)**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Client</td>
<td>The number of requests in which the RADIUS client could not be identified.</td>
</tr>
<tr>
<td></td>
<td><em>A device might be configured to use Steel-Belted Radius but no RADIUS client entry has been created with the name and/or IP address of the client; or the RADIUS client entry might be configured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.</em></td>
</tr>
<tr>
<td>Invalid Shared Secret</td>
<td>The number of packets for which an incorrect digital signature was received.</td>
</tr>
<tr>
<td></td>
<td><em>The shared secret does not match between Steel-Belted Radius and the client device; or some rogue device is attempting to compromise RADIUS security.</em></td>
</tr>
<tr>
<td>Insufficient Resources</td>
<td>The number of rejects due to a server resource problem.</td>
</tr>
<tr>
<td>Proxy Failure</td>
<td>The number of times that proxy RADIUS forwarding failed.</td>
</tr>
<tr>
<td>Retries Received</td>
<td></td>
</tr>
<tr>
<td>Transactions Retried</td>
<td>The number of requests for which one or more duplicates was received.</td>
</tr>
<tr>
<td>Total Retry Packets</td>
<td>The number of duplicate packets received.</td>
</tr>
<tr>
<td>Interim Requests</td>
<td>The number of interim accounting packets received.</td>
</tr>
</tbody>
</table>

**Displaying Proxied Request Statistics**

To display proxied request statistics, open the Statistics panel, click the System tab, pull down the View list, and choose Proxied Requests.

### Figure 118: Statistics Panel: Proxied Request Statistics
Table 34 describes the proxy request statistics, with possible interpretations in italics.

**Table 34: Proxy Statistics**

<table>
<thead>
<tr>
<th>Proxy Statistic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions</td>
<td>The number of authentication and accounting transaction totals.</td>
</tr>
<tr>
<td>Authentication</td>
<td>The number of authentication transactions between the proxy and target RADIUS servers.</td>
</tr>
<tr>
<td>Accounting</td>
<td>The number of accounting transactions between the proxy and target RADIUS servers.</td>
</tr>
<tr>
<td>Total Transactions</td>
<td>The sum of the authentication and accounting transaction totals.</td>
</tr>
<tr>
<td>Failure Details</td>
<td></td>
</tr>
<tr>
<td>Timed Out</td>
<td>The number of RADIUS transactions that timed out. This means that after all retry attempts were made, the transaction still timed out.</td>
</tr>
<tr>
<td>Invalid Response</td>
<td>The number of invalid RADIUS responses received.</td>
</tr>
<tr>
<td></td>
<td><em>A target is sending incorrectly formed packets to Steel-Belted Radius; there is a configuration error, the target RADIUS server does not conform to the RADIUS standard, or Steel-Belted Radius did not receive a proxy state echo in the received packet.</em></td>
</tr>
<tr>
<td>Invalid Shared Secret</td>
<td>The number of packets for which an incorrect digital signature was received.</td>
</tr>
<tr>
<td></td>
<td><em>The shared secret does not match between Steel-Belted Radius and the target; or some unauthorized rogue device is attempting to compromise RADIUS security.</em></td>
</tr>
<tr>
<td>Insufficient Resources</td>
<td>The number of rejects due to a server resource problem.</td>
</tr>
<tr>
<td>Retries Sent</td>
<td></td>
</tr>
<tr>
<td>Transactions Retried</td>
<td>The number of requests for which one or more retried transmissions was performed.</td>
</tr>
<tr>
<td>Total Retry Packets</td>
<td>The number of duplicate packets received.</td>
</tr>
</tbody>
</table>

**Displaying RADIUS Client Statistics**

RADIUS client statistics provide information about the number of authentication and accounting requests by client.
To display statistics for RADIUS clients:

1. Open the Statistics panel and click the RADIUS Clients tab.

2. Use the View list to display the type of statistics you want to display.
   - **Summary**—Displays the number of authentication requests, Access-Accepts, and Accept-Reject messages and the total number of accounting requests, starts, and stops for each RADIUS client.
   - **Authentication Request Details**—Displays the number of duplicate messages, challenges, messages containing invalid authentication information, bad authentication requests, bad types, and dropped requests for each RADIUS client.
   - **Accounting Request Types**—Displays the number of accounting start messages, accounting stop messages, interim messages, Accounting-On messages, Accounting-Off messages, and acknowledgement messages sent for each RADIUS client.
   - **Accounting Request Diagnostics**—Displays the number of duplicate messages, messages with invalid secrets, malformed messages, messages with incorrect types, ignored messages, and dropped requests for each RADIUS client.

3. Optionally, sort the messages by clicking a column header.
Displaying RADIUS Proxy Targets Statistics

RADIUS proxy target statistics provide information about the number of authentication and accounting transactions associated with each proxy target.

**Figure 120: Statistics Tab: RADIUS Proxy Targets Tab**

To display statistics for RADIUS proxy targets:

1. Open the Statistics panel and click the **RADIUS Proxy Targets** tab.
2. Use the **View** list to display the type of statistics you want to display.
   - **Summary**—Displays the number of authentication requests, accepts and reject messages, and the number of accounting requests and responses for each RADIUS proxy target.
   - **Authentication Request Details**—Displays the number of outstanding messages, retransmitted messages, and challenges, along with the most recent response time for the proxy target.
   - **Authentication Request Diagnostics**—Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.
   - **Accounting Request Types**—Displays the number of outstanding messages and retransmitted messages, along with the most recent response time for the proxy target.
   - **Accounting Request Diagnostics**—Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.
3. Optionally, sort the messages by clicking a column header.
Displaying IP Address Pool Statistics

IP address pool statistics provide a summary of the number of addresses allocated from each IPv4 address pool and how many addresses remain available.

Figure 121: Statistics Panel: IP Address Pools Tab
Chapter 19
Logging and Reporting

This chapter describes how to set up and use logging and reporting functions in Steel-Belted Radius.

Logging Files

The following files establish settings for logging and reporting. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>account.ini</td>
<td>Controls how RADIUS accounting attributes are logged.</td>
</tr>
<tr>
<td>authlog.ini</td>
<td>Controls how RADIUS authentication requests are logged by Steel-Belted Radius.</td>
</tr>
<tr>
<td>authReport.ini</td>
<td>Controls what authentication logs Steel-Belted Radius generates.</td>
</tr>
<tr>
<td>authReportAccept.ini</td>
<td>Controls options for the acceptance authentication log file.</td>
</tr>
<tr>
<td>authReportReject.ini</td>
<td>Controls options for the rejection authentication log file.</td>
</tr>
<tr>
<td>authReportUnknownClient.ini</td>
<td>Controls options for the unknown client authentication log file.</td>
</tr>
<tr>
<td>events.ini</td>
<td>Controls dilutions and thresholds for Steel-Belted Radius events used to communicate failures, warnings, and other information.</td>
</tr>
<tr>
<td>radius.ini</td>
<td>Controls (among other things) the types of messages Steel-Belted Radius records in the server log file and the location of the log directory.</td>
</tr>
</tbody>
</table>

Displaying the Current Sessions List

Steel-Belted Radius tracks the status of the user connections that it authenticates. To obtain a real-time snapshot of currently active connections, display the Current Sessions list. Because the Current Sessions list is based on RADIUS accounting data, the list is accurate only if all of your network access devices are configured to support RADIUS accounting.
Each server has its own Current Sessions display. Therefore, when you view this display, it typically reflects only the activity on the Steel-Belted Radius server to which you are connected. The Current Users display on a specific server reflects the activity across your entire RADIUS configuration only if (1) all clients in your configuration support RADIUS accounting, and (2) all clients are configured to send accounting messages to the server you are viewing.

NOTE: Steel-Belted Radius maintains the Current User list on disk. The information is preserved if you unload and reload the server.

To display the Current Sessions list (Figure 122), open the Reports panel and click the Current Sessions tab.

Figure 122: Reports Panel: Current Sessions List

Table 36 describes the fields for each active session in the Current Sessions list.

Table 36: Sessions List Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>The identifier for the RADIUS client, which is typically the name or IP address of the device.</td>
</tr>
<tr>
<td>Port</td>
<td>The UDP port number on the RADIUS client that has been assigned to the connection. To determine slot number of the physical port on the RADIUS client, consult the device documentation.</td>
</tr>
<tr>
<td>Port Type</td>
<td>Describes how the port is used or configured.</td>
</tr>
<tr>
<td>Time</td>
<td>Identifies the date and time at which the connection was opened.</td>
</tr>
<tr>
<td>Session ID</td>
<td>Identifies the session key, which is a number generated by the RADIUS client.</td>
</tr>
</tbody>
</table>
Searching the Current Sessions List

You can search the Current Sessions list to display only those sessions that match user name or client name criteria.

To search the Current Sessions list:

1. Open the Reports panel and click the Current Sessions tab.
2. Click the Search button.

The Search Current Sessions dialog (Figure 123) opens.

3. Enter the search criteria you want to use.
   - To search for a specific username, enter it in the User name field.
   - To search for a specific RADIUS client, enter a client name in the Client name field.
4. Click Search.

Table 36: Sessions List Fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Displays the name of the authenticated user.</td>
</tr>
<tr>
<td></td>
<td>- If the user is local (native), the field shows only the username, in the form username</td>
</tr>
<tr>
<td></td>
<td>- If the user is non-local, the field shows the remote system name as well as the username, in the form \systemname\username.</td>
</tr>
<tr>
<td></td>
<td>- If the user is associated with a specific tunnel, the field shows the tunnel name as well as the username, in the form \tunnelname\username.</td>
</tr>
<tr>
<td>IP Address</td>
<td>Identifies the IP address that was assigned to the user from an IP address pool. This field will be blank if a static IP address was assigned.</td>
</tr>
</tbody>
</table>

NOTE: For tunnel connections, if Steel-Belted Radius was used to authenticate both the user and the tunnel, then two entries are displayed in the Current Sessions panel: one entry for the authenticated user, and one for the authenticated tunnel.
Deleting Entries from the Sessions List

Normally, the system maintains the information in the Sessions list based on accounting information received from the RADIUS client. However, a user who has logged off may still be identified as active in the Current Sessions list if communications between the RADIUS client and Steel-Belted Radius fail or if either the RADIUS client or Steel-Belted Radius is taken down for a period of time.

In most cases, Steel-Belted Radius can correct such anomalies itself. For example, if a new user dials in to the same port on the same RADIUS client, Steel-Belted Radius infers that the previous user must have disconnected and removes the entry.

You can manually correct the Sessions list by highlighting any entry and clicking Delete. This removes the user from the list and decrements the user’s connection count (if it is being tracked) by one. Any pooled IP or IPX address assigned to the deleted user is returned to the appropriate pool.

Displaying the Authentication Log Files

The Auth Logs tab (Figure 124) on the Reports panel lets you enable and display the following authentication log files:

- **Successful request authentication log file**—The successful request authentication log file identifies the authentication requests that were approved by Steel-Belted Radius.

- **Invalid shared secrets authentication log file**—The invalid shared secrets authentication log file identifies the authentication requests that failed because a known RADIUS client supplied an incorrect shared secret. This condition is detectable only if the authentication request contains a Message-Authenticator attribute, which is required if credentials are of an EAP type but optional if credentials are PAP, CHAP, or MS-CHAP.

- **Failed request authentication log file**—The failed request authentication log file identifies the authentication requests that were rejected because the user supplied incorrect credentials.

- **Unknown client request authentication log file**—The unknown client request authentication log file identifies authentication requests received from unknown RADIUS clients.

File Permissions for Log Files (Solaris/Linux)

When you run Steel-Belted Radius on a Solaris or Linux server, you can specify the users who are authorized to read or edit important files, such as authentication and accounting log files. For example, you can specify that system administrators who install and configure Steel-Belted Radius have read/write access for system log files and that network operators who monitor Steel-Belted Radius have read-only (or no) access for system log files.
Security Groups and Permissions

Each file and directory on a Solaris or Linux server has three security groups associated with it:

- The Owner identifies the person who created or owns the file.
- The Group security group identifies the set of users who are members of the group or groups to which the file Owner belongs. Group members can exercise special privileges with respect to that file. A user can belong to more than one group.
- The Other security group consists of the set of all users who do not belong to Owner or Group.

Each security group has three flags that control what privileges that group can exercise with respect to the file or directory:

- The Read flag (r) determines whether the file can be read. The Read flag has an octal value of 4.
- The Write flag (w) determines whether the security group can create, modify, or delete the file. The Write flag has an octal value of 2.
- The Execute flag (x) determines whether the security group can run a script or executable file. The Execute flag has an octal value of 1.

For example, a file owner might have rwx permission for a file, which indicates the file owner has read/write/execute access to the file. Similarly, Other might have r- permission (where - indicates no permission), which means that the user can read but not edit or execute the file.

You can add the octal values for permission flags to generate a numeric representation of the file permissions for Owner, Group, and Other:

- 1 = execute only
- 2 = write only
- 3 = write and execute (1 + 2)
- 4 = read only
- 5 = read and execute (4 + 1)
- 6 = read and write (4 + 2)
- 7 = read and write and execute (4 + 2 + 1)
The security permissions exercised by Owner/Group/Other are typically expressed as string or a three-digit number. Table 37 provides examples of different file permissions.

### Table 37: File Permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Octal value</th>
<th>What It Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>-rwxrwxrwx</td>
<td>777</td>
<td>Read, write, and executable for Owner/Group/Other</td>
</tr>
<tr>
<td>-rw-rw-r--</td>
<td>664</td>
<td>Read and write for Owner/Group, read access for Other</td>
</tr>
<tr>
<td>-rw-rw-r-</td>
<td>660</td>
<td>Read and write for Owner/Group, no access for Other</td>
</tr>
<tr>
<td>-rwx------</td>
<td>700</td>
<td>Read, write, and executable for owner only</td>
</tr>
<tr>
<td>-rw-rw-rw</td>
<td>666</td>
<td>Read and write for owner, group, and all others</td>
</tr>
</tbody>
</table>

The UNIX `chown` command lets you change the owner or group (or both) associated with a file or directory. The UNIX `chmod` command lets you change the permissions of files and directories.

### Using the User File Creation Mode Mask

The user file mode creation mode mask (often abbreviated as `umask`) determines the default file system mode for newly created files of the current process. Solaris/Linux hosts typically have a hierarchy of `umask` values: a server-level `umask` value, which can be overridden by a user-, shell-, or application-level `umask` value. The result is an ambient `umask` value, which determines what file permissions are used when files are created by any given process.

The `umask` value is a three-digit octal number. The first digit sets the mask for Owner, the second for Group, and the third for Other. The `umask` value identifies the permissions that are withheld when a file is created: the `umask` value is subtracted from the full access mode value (777) to determine the access permissions for a new file. For example, if the `umask` value for a process is set to 022, the Write permission for Group and Other are withheld from the full access mode value (777), resulting in a file permission of 755 (rwxr-xr-x). Similarly, if the `umask` value of 177 is configured for a process (explicitly or by virtue of the ambient `umask`), files created by the process have a file permission of 600 (rw--). Table 38 summarizes the result of using different octal numbers in a `umask` value.

### Table 38: Summary of umask Permissions

<table>
<thead>
<tr>
<th>Octal Number</th>
<th>Access</th>
<th>Permission Resulting From umask Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rwx</td>
<td>Read, Write, Execute</td>
</tr>
<tr>
<td>1</td>
<td>rw-</td>
<td>Read, Write</td>
</tr>
<tr>
<td>2</td>
<td>r-x</td>
<td>Read, Execute only</td>
</tr>
<tr>
<td>3</td>
<td>r--</td>
<td>Read only</td>
</tr>
<tr>
<td>4</td>
<td>-wx</td>
<td>Write, Execute only</td>
</tr>
<tr>
<td>5</td>
<td>-w-</td>
<td>Write only</td>
</tr>
<tr>
<td>6</td>
<td>--x</td>
<td>Execute only</td>
</tr>
<tr>
<td>7</td>
<td>---</td>
<td>No permissions</td>
</tr>
</tbody>
</table>
The `umask` value affects a file’s access permissions only when the file is created. If you change the `umask` value, access permissions for existing files are not affected. Similarly, you can use the `chown` and `chmod` commands to change a file’s access permissions after the file has been created.

**Implementing Default File Permissions in Steel-Belted Radius**

The `RADIUSMASK` parameter in the `sbrd.conf` file specifies the application-level `umask` value used to establish access permissions for all files created by Steel-Belted Radius. Refer to the *Steel-Belted Radius Reference Guide* for information on configuring the `sbrd.conf` file.

If you do not specify a value for the `RADIUSMASK` parameter, Steel-Belted Radius uses the ambient `umask` value established by the server-, user- or shell-level `umask` value to determine the access permissions for files it creates.

Some log files have explicit controls that let you override the `umask` value established by the `RADIUSMASK` parameter or the ambient `umask` value. See **Implementing Override File Permissions in Steel-Belted Radius** for more information on overriding the application-level default `umask` value.

As previously noted, the `umask` value affects a file’s access permissions only when the file is created. If you change the `RADIUSMASK` setting, new files created by Steel-Belted Radius are assigned the access permission specified by the new setting. This includes files that roll over periodically; the existing file would retain the access file permission it received when it was created, and the new file would be assigned the access permission specified by the new `RADIUSMASK` value.

**NOTE:** The Execute file permission value for files created by Steel-Belted Radius is always set to None for Owner, Group, and Other. Thus, a `umask` value of 0 (no restrictions) is equivalent to a `umask` value of 1 (read/write permission) for files created by Steel-Belted Radius.

**Implementing Override File Permissions in Steel-Belted Radius**

To override file permissions established by the Steel-Belted Radius `RADIUSMASK` or the ambient `umask` for specific log files, you must modify the `LogFilePermissions` parameter in the applicable initialization (.ini) file.

Table 39 identifies the configuration file you must modify to configure non-default file permissions for Steel-Belted Radius log files.

**Table 39: Configuration Files for Setting Log File Permissions**

<table>
<thead>
<tr>
<th>Controlled Files</th>
<th>Configuration File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Diagnostics log (RADIUS log)</td>
<td><code>radius.ini</code></td>
</tr>
<tr>
<td>Authentication Reporting Library accepts log</td>
<td><code>authReportAccept.ini</code></td>
</tr>
<tr>
<td>Authentication Reporting Library bad shared secret log</td>
<td><code>authReportBadSharedSecret.ini</code></td>
</tr>
<tr>
<td>Authentication Reporting Library rejects log</td>
<td><code>authReportReject.ini</code></td>
</tr>
<tr>
<td>Authentication Reporting Library unknown client log</td>
<td><code>authReportUnknownClient.ini</code></td>
</tr>
</tbody>
</table>
Table 39: Configuration Files for Setting Log File Permissions (continued)

<table>
<thead>
<tr>
<th>Controlled Files</th>
<th>Configuration File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Logging Library logs and header check-point logs</td>
<td>authlog.ini</td>
</tr>
<tr>
<td>Accounting Library logs and header check-point logs</td>
<td>account.ini</td>
</tr>
<tr>
<td>Server Statistics logs and header check-point logs</td>
<td>statlog.ini</td>
</tr>
</tbody>
</table>

The syntax for the LogFilePermissions parameter is:

\[
\text{LogFilePermissions} = \text{owner:group mode}
\]

- Specify the owner and group settings by entering character strings or decimal integers, as used for arguments to the UNIX chown(1) command. For example, ralphw:proj, ralphw:120, or 1007:120.

- Specify the mode setting as a character string or an octal integer. When permissions are specified as a character string, they follow the format that is used by the UNIX ls(1) command; for example, rwxrwxrwx. When permissions are specified as an octal integer, they follow the format used for arguments to the UNIX chmod(1) command; for example, 666.

**NOTE:** You can specify only read/write permissions for a Steel-Belted Radius file. You cannot specify execute permissions for Steel-Belted Radius files.

The value of each LogFilePermissions parameter is read when the Steel-Belted Radius server is started or restarted.

- If you enter a valid value for a LogFilePermissions parameter, the ownership and permissions of the controlled log file are set as specified whenever the file is opened or created.

- If you do not enter a value for a LogFilePermissions parameter, the ownership and permissions of the controlled file are not changed. The controlled file is created using the ownership of the account that is executing the server and the permissions that are derived from the default RADIUSMASK value or from the ambient umask setting. If the file already exists, new information is appended without changing the existing ownership and permissions of the controlled file.

- If you enter an invalid value for a LogFilePermissions setting, then the ownership of the controlled log file defaults to the effective user/group ID of the server process (normally root:other on Solaris and root:root on Linux), and the permissions for the controlled file default to 0600 (rwx—–). This ensures that the affected log file can always be opened without any escalation of file access privileges. Messages similar to the following are logged whenever an explicit file access control is misconfigured:

```
Invalid LogfilePermissions specified in radius.ini [Configuration]: -rwx—–
Server log file permissions defaulted to 0:0 0600
```
Enabling and Disabling the Authentication Log Files

To enable an authentication log file:

1. Open the Reports panel and click the Auth Logs tab.

Figure 124: Reports Panel: Auth Logs Tab

2. Use the Logs list to select the authentication log file you want to enable or disable.

3. Click the Enable logging check box to enable the specified authentication log.

Clear the Enable logging check box to disable the specified authentication log.

Viewing the Authentication Log Files

To display an authentication log file:

1. Open the Reports panel and click the Auth Logs tab.

2. Use the Logs list to select the log file you want to display.

3. Click the View button.

The Log List dialog (Figure 125) opens.
Displaying the Authentication Log Files

4. Select the log you want to display and click View.

By default, SBR Administrator displays the authentication log file 20 lines at a time. To change the number of lines displayed, enter a different number in the Page size field before you click View.

5. When the authentication log file dialog opens, click the Up and Down arrows to page through the log file.

To sort the authentication log file, click the appropriate column header.

To print the authentication log file, click the Print button.

To refresh the authentication log file display, click the Refresh button.

6. When you are finished, click Close.

Saving the Log Files

To save an authentication log file to a text file:

1. Open the Reports panel and click the Auth Logs tab.

2. Use the Logs list to select the authentication log file you want to save.

3. Click the View button.

The Log List dialog (Figure 125) opens.

4. Select the log you want to save and click Save.

The Save Log File As dialog (Figure 126) opens.
5. Specify the name and destination for the log file and click **Save**.

**Searching the Log Files**

You can search the Steel-Belted Radius authentication log files to display messages within a specified time range, messages relating to a specific client, or messages relating to a specific user.

To search the authentication log files:

1. Open the Reports panel and click the **Auth Logs** tab.
2. Use the **Logs** list to select the type of authentication log file you want to search.
3. Click the **Search** button.

The Search Logs dialog (Figure 127) opens.

**Figure 127: Search Logs Dialog**
4. If you want to search the authentication log file for messages within a specified time range:
   a. Specify the starting date/time in the range by clicking the Now radio button or by clicking the From: Specific date radio button.
   b. Specify the ending date/time in the range by clicking the No limit radio button or by clicking the To: Specified date radio button and entering a date and time.

5. If you want to filter message so that you see only those relating to a specified RADIUS client, click the RADIUS client check box and enter the name of the RADIUS client in the RADIUS client text field.

6. If you want to filter message so that you see only those relating to a specified user, click the User name check box and enter the name of the user in the User name text field.

7. If you want to limit the number of messages you want SBR Administrator to display, enter a number in the Maximum returns field.

8. Click OK.

---

**Configuring the Log Retention Period**

Each day at midnight, the previous day’s log files are completed, and new log files are created for the new day’s transactions. To prevent the log files from filling up available disk space, you can configure Steel-Belted Radius to discard the log files after a specified number of days.

To configure the log retention period:

1. Open the Reports panel and click the Settings tab.

2. When the Settings tab (Figure 128) opens, enter the number of days you want Steel-Belted Radius to retain log file in the Days to keep server log file field.
Using the Server Log File

The server log file records RADIUS events, such as server startup or shutdown or user authentication or rejection, as a series of messages in an ASCII text file. Each line of the server log file identifies the date and time of the RADIUS event, followed by event details. You can open the current log file while Steel-Belted Radius is running.

Server log files are kept for the number of days specified in the Settings tab in the Reports panel (described in “Configuring the Log Retention Period” on page 250) and then deleted to conserve disk space.

Optionally, you can specify a maximum size for a server log file by entering a non-zero value for the LogfileMaxMBytes setting in the [Configuration] section of the radius.ini file.

- If a maximum file size is set, the server log filename identifies the date and time it was opened (YYYYMMDD_HHMM.log). When the current server log file approaches the specified number of megabytes (1024 x 1024 bytes), the current log file is closed and a new one is opened. The closed file will be slightly smaller than the specified maximum file size.

- If the maximum file size is set to 0 (or if the LogfileMaxMBytes setting is absent), the server log file size is ignored and log file names are datestamped to identify when they were opened (YYYYMMDD.log).

**NOTE:** The size of the log file is checked once per minute, and the log file cannot roll over more than once a minute. The log file may exceed the specified maximum file size temporarily (for less than a minute) after it passes the LogfileMaxMBytes threshold between size checks.

By default, server log files are located in the RADIUS database directory. You can specify an alternate destination directory in the [Configuration] section of the radius.ini file.
Level of Logging Detail

You can control the level of detail recorded in server log files by use of the LogLevel, LogAccept, and LogReject settings.

The LogLevel setting determines the level of detail given in the server log file. The LogLevel can be the number 0, 1, or 2, where 0 is the least amount of information, 1 is intermediate, and 2 is the most verbose. The LogLevel setting is specified in the [Configuration] section of radius.ini and in the [Settings] sections of .aut files.

The LogAccept and LogReject flags allow you to turn on or off the logging of Access-Accept and Access-Reject messages in the server log file. These flags are set in the [Configuration] section of radius.ini: a value of 1 causes these messages to be logged, and a value of 0 causes the messages to be omitted. An Accept or Reject is logged only if LogAccept or LogReject, respectively, is enabled and the LogLevel is “verbose” enough for the message to be recorded.

The TraceLevel setting specifies whether packets should be logged when they are received and being processed, and what level of detail should be recorded in the log.

Using the Accounting Log File

RADIUS accounting events are recorded in the accounting log file. Accounting events include START messages, which indicate the beginning of a connection; STOP messages, which indicate the termination of a connection; and INTERIM messages, which indicate a connection is ongoing.

Accounting log files use comma-delimited, ASCII format, and are intended for import into a spreadsheet or database program. Accounting log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of the account.ini file. Accounting log files are named yyyymmdd.ACT, where yyyy is the 4-digit year, mm is the month, and dd is the day on which the log file was created.

Accounting log files are kept for the number of days specified in the Settings tab of the Reports panel, and are deleted after that to conserve disk space.

The current log file can be opened while Steel-Belted Radius is running.

Accounting Log File Format

The first six fields in every accounting log entry are provided by Steel-Belted Radius for your convenience in reading and sorting the file:

- Date—The date when the event occurred
- Time—The time when the event occurred
- RAS-Client—The name or IP address of the RADIUS client sending the accounting record
Using the Accounting Log File

Chapter 19: Logging and Reporting

- **Record-Type**—START, STOP, INTERIM, ON, or OFF, the standard RADIUS accounting packet types
- **Full-Name**—The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- **Auth-Type**—A number that indicates the class of authentication performed:
  
  
  0—Native  
  10—SecurID User  
  11—SecurID Prefix  
  12—SecurID Suffix  
  13—Solaris User  
  14—Solaris Group  
  15—TACACS+ User  
  16—TACACS+ Prefix  
  17—TACACS+ Suffix  
  100—Tunnel User  
  200—External Database  
  (other)—Proxy

By default, the standard RADIUS attributes follow the **Auth-Type** identifier. See "Standard RADIUS Accounting Attributes" on page 254.

You can include vendor-specific attributes if the device sending the accounting packet supports them. For more information, see “Vendor-Specific Attributes” on page 10.

You can edit the account.ini initialization file to add, remove or reorder the standard RADIUS or vendor-specific attributes that are logged. For information on account.ini, refer to the Steel-Belted Radius Reference Guide.

**First Line Headings**

The first line of the accounting log file is a file header that lists the attributes that have been enabled for logging in the order in which they are logged. The following example of a first line shows required headings in bold italic, standard RADIUS headings in bold, and vendor-specific headings in regular text:

"Date","Time","RAS-Client","Record-Type","Full-Name","Auth-Type","User-Name", "NAS-Port","Acct-Status-Type","Acct-Delay-Time","Acct-Input-Octets", "Acct-Output-Octets","Acct-Session-Id","Acct-Authentic","Acct-Session-Time", "Acct-Input-Packets","Acct-Output-Packets","Acct-Termination-Cause", "Acct-Multi-Session-Id","Acct-Link-Count","Acc-Err-Message", "Nautica-Acct-SessionId","Nautica-Acct-Direction", "Nautica-Acct-CauseProtocol","Nautica-Acct-CauseSource", "Telebit-Accounting-Info","Last-Number-Dialed-Out", "Last-Number-Dialed-In-DNIS","Last-Callers-Number-ANI", "Channel","Event-Id","Event-Date-Time","Call-Start-Date-Time","Call-End-Date-Time", "Default-DTE-Data-Rate","Initial-Rx-Link-Data-Rate", "Initial-Tx-Link-Data-Rate","Initial-Tx-Link-Data-Rate", "Final-Rx-Link-Data-Rate","Initial-Tx-Link-Data-Rate", "Final-Tx-Link-Data-Rate","Sync-Async-Mode", "Originate-Answer-Mode","Modulation-Type", "Equalization-Type","Fallback-Enabled","Characters-Sent", "Characters-Received","Blocks-Sent","Blocks-Received", "Blocks-Resent","Retrains-Requested","Retrains-Granted", "..."
"Line-Reversals", "Number-Of-Characters-Lost",
"Number-of-Blers", "Number-of-Link-Timeouts",
"Number-of-Fallbacks", "Number-of-Upshifts",
"Number-of-Link-NAKs", "Back-Channel-Data-Rate",
"Simplified-MNP-Levels", "Simplified-V42bis-Usage", "PW_VPN_ID"

**Comma Placeholders**

Steel-Belted Radius writes accounting events to the accounting log file. If an event recorded in the accounting log file does not have data for every attribute, a comma “placeholder” marks the empty entry, so that all entries remain correctly aligned with their headings. For example, based on the “first line” of headings described above, the following is a valid accounting log entry, in which the value of the Acct-Status-Type attribute is 7:

"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "

**Standard RADIUS Accounting Attributes**

Table 40 lists the standard RADIUS accounting attributes defined in RFC 2866, “RADIUS Accounting.”

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-Name</td>
<td>The name of the user as received by the client.</td>
</tr>
<tr>
<td>NAS-Port</td>
<td>The port number on the client device.</td>
</tr>
<tr>
<td>Acct-Status-Type</td>
<td>A number that indicates the beginning or ending of the user service:</td>
</tr>
<tr>
<td></td>
<td>1—Start</td>
</tr>
<tr>
<td></td>
<td>2—Stop</td>
</tr>
<tr>
<td></td>
<td>3—Interim-Acct</td>
</tr>
<tr>
<td></td>
<td>7—Accounting-On</td>
</tr>
<tr>
<td></td>
<td>8—Accounting-Off</td>
</tr>
<tr>
<td>Acct-Delay-Time</td>
<td>Indicates how many seconds the client has been trying to send this record, which can be subtracted from the time of arrival on the server to find the approximate time of the event generating this request.</td>
</tr>
<tr>
<td>Acct-Input-Octets</td>
<td>Number of octets (bytes) received by the port over the connection; present only in STOP records.</td>
</tr>
<tr>
<td>Acct-Output-Octets</td>
<td>Number of octets (bytes) sent by the port over the connection; present only in STOP records.</td>
</tr>
<tr>
<td>Acct-Session-Id</td>
<td>Identifier used to match START and STOP records in a log file.</td>
</tr>
<tr>
<td>Acct-Authentic</td>
<td>Indicates how the user was authenticated by RADIUS, the network access device (local), or another remote authentication protocol:</td>
</tr>
<tr>
<td></td>
<td>1—RADIUS</td>
</tr>
<tr>
<td></td>
<td>2—Local</td>
</tr>
<tr>
<td></td>
<td>3—Remote</td>
</tr>
<tr>
<td>Acct-Session-Time</td>
<td>Elapsed time of connection in seconds; present only in STOP records.</td>
</tr>
<tr>
<td>Acct-Input-Packets</td>
<td>Number of packets received by the port over the connection; present only in STOP records.</td>
</tr>
</tbody>
</table>
### Table 40: Standard RADIUS Accounting Attributes (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acct-Output-Packets</td>
<td>Number of packets sent by the port over the connection; present only in STOP records.</td>
</tr>
<tr>
<td>Acct-Termination-Cause</td>
<td>Number that indicates how the session was terminated; present only in STOP records:</td>
</tr>
<tr>
<td></td>
<td>1—User Request</td>
</tr>
<tr>
<td></td>
<td>2—Lost Carrier</td>
</tr>
<tr>
<td></td>
<td>3—Lost Service</td>
</tr>
<tr>
<td></td>
<td>4—Idle Timeout</td>
</tr>
<tr>
<td></td>
<td>5—Session Timeout</td>
</tr>
<tr>
<td></td>
<td>6—Admin Reset</td>
</tr>
<tr>
<td></td>
<td>7—Admin Reboot</td>
</tr>
<tr>
<td></td>
<td>8—Port Error</td>
</tr>
<tr>
<td></td>
<td>9—NAS Error</td>
</tr>
<tr>
<td></td>
<td>10—NAS Request</td>
</tr>
<tr>
<td></td>
<td>11—NAS Reboot</td>
</tr>
<tr>
<td></td>
<td>12—Port Unneeded</td>
</tr>
<tr>
<td></td>
<td>13—Port Preempted</td>
</tr>
<tr>
<td></td>
<td>14—Port Suspended</td>
</tr>
<tr>
<td></td>
<td>15—Service Unavailable</td>
</tr>
<tr>
<td></td>
<td>16—Callback</td>
</tr>
<tr>
<td></td>
<td>17—User Error</td>
</tr>
<tr>
<td></td>
<td>18—Host Request</td>
</tr>
<tr>
<td>Acct-Multi-Session-Id</td>
<td>Unique accounting identifier to make it easy to link together multiple related sessions in a log file.</td>
</tr>
<tr>
<td>Acct-Link-Count</td>
<td>The count of links that are known to have been in a given multi-link session at the time the accounting record is generated.</td>
</tr>
</tbody>
</table>
# Appendix A
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.1X</td>
<td>The IEEE 802.1X standard defines a mechanism that allows a supplicant (client) to connect to a wireless access point or wired switch (authenticator) so that the supplicant can provide authentication credentials that can be verified by an authentication server.</td>
</tr>
<tr>
<td>AAA</td>
<td>Authentication, authorization, and accounting.</td>
</tr>
<tr>
<td>accounting</td>
<td>The process of recording and aggregating resource use statistics and log files for a user, connection session, or function for billing, system diagnosis, and usage planning.</td>
</tr>
<tr>
<td>agent</td>
<td>SNMP module on a managed device that responds to requests from a management station and sends traps to one or more recipients (trap sinks) to inform administrators of potential problems.</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point. A device that serves as a communication hub to connect 802.1X wireless clients to a wired network.</td>
</tr>
<tr>
<td>attribute</td>
<td>RADIUS attributes carry the specific authentication, authorization, and accounting.</td>
</tr>
<tr>
<td>authentication</td>
<td>The process of verifying the identity of a person or file system and whether the person is allowed on a protected network.</td>
</tr>
<tr>
<td>authentication server</td>
<td>A back-end database server that verifies, from the credentials provided by an access client, whether the access client is authorized to use network resources.</td>
</tr>
<tr>
<td>authorization</td>
<td>The process of controlling the access settings, such as privileges and time limits, that the user can exercise on a protected network.</td>
</tr>
<tr>
<td>AVP</td>
<td>Attribute-value pair. An attribute and its corresponding value.; for example, User-Name = admin.</td>
</tr>
<tr>
<td>blacklist</td>
<td>A profile of checklist attributes that cause Steel-Belted Radius to reject an authentication request. For example, a blacklist profile might specify calling station phone numbers or IP addresses that are blocked by Steel-Belted Radius.</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate authority. A trusted entity that registers the digital identity of a site or individual and issues a digital certificate that guarantees the binding between the identity and the data items in a certificate.</td>
</tr>
</tbody>
</table>
CCM  Centralized configuration management. The process by which information is shared between a primary RADIUS server and one or more replica RADIUS servers in a multi-server environment.

certificate  A digital file signed by a CA that guarantees the binding between an identity and the contents of the certificate.

CHAP  Challenge Handshake Authentication Protocol. An authentication protocol where a server sends a challenge to a requestor after a link has been established. The requestor responds with a value obtained by executing a hash function. The server verifies the response by calculating its own hash value: if the two hash values match, the authentication is acknowledged.

checklist  A list of attributes that must accompany a request for connection before the connection request can be authenticated.

CIDR  Classless Inter-Domain Routing. In CIDR notation, an IP address is represented as A.B.C.D/n, where /n identifies the IP prefix or network prefix. The IP prefix identifies the number of significant bits used to identify a network. For example, 192.168.1.22/18 means “use the first 18 bits to represent the network and the remaining 14 bits to identify hosts.” Common prefixes are /8 (Class A network), /16 (Class B network), /24 (Class C network), and /32.

community  An SNMP community is a group of devices and management stations running SNMP. An SNMP device or agent may belong to more than one SNMP community.

community string  Character string included in SNMP messages to identify valid sources for SNMP requests and to limit access to authorized devices.
- The read community string allows an SNMP management station to issue Get and GetNext messages.
- The write community string allows an SNMP management station to issue Set messages.

credentials  Data that is verified when presented to an authenticator, such as a password or a digital certificate.

CRL  Certificate Revocation List. A data structure that identifies the digital certificates that have been invalidated by the certificates’ issuing CA prior to their expiration date.

daemon  A program on a Solaris or Linux host that runs continuously to handle service requests.

dictionary  Text file that maps the attribute/value pairs supported by third-party RADIUS vendors.

DHCP  Dynamic Host Configuration Protocol. Protocol by which a server automatically assigns (leases) a network address and other configuration settings to a client temporarily or permanently.

DNIS  Dialed number identification service. A telephone service that identifies what number was dialed by a caller.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS</td>
<td>Domain Name Service. Internet protocol for mapping host names, domain names, and aliases to IP addresses.</td>
</tr>
<tr>
<td>EAP-15</td>
<td>See POTP.</td>
</tr>
<tr>
<td>EAP-FAST</td>
<td>Authentication method that uses EAP (Extensible Authentication Protocol) and FAST (Flexible Authentication via Secure Tunneling).</td>
</tr>
<tr>
<td>EAP-TLS</td>
<td>Authentication method that uses EAP (Extensible Authentication Protocol) and TLS (Transport Layer Security).</td>
</tr>
<tr>
<td>EAP-TTLS</td>
<td>Authentication method that uses EAP (Extensible Authentication Protocol) and TTLS (Tunneled Transport Layer Security).</td>
</tr>
<tr>
<td>GTC</td>
<td>Generic Token Card.</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers.</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force. Technical subdivision of the Internet Architecture Board that coordinates the development of Internet standards.</td>
</tr>
<tr>
<td>IPv4</td>
<td>Implementation of the TCP/IP suite that uses a 32-bit addressing structure.</td>
</tr>
<tr>
<td>IPv6</td>
<td>Implementation of the TCP/IP suite that uses a 128-bit addressing structure.</td>
</tr>
<tr>
<td>Java</td>
<td>Programming language designed for use in distributed environments such as the Internet.</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity. Application programming interface for accessing a database from programs written in Java.</td>
</tr>
<tr>
<td>LCI</td>
<td>LDAP configuration interface.</td>
</tr>
<tr>
<td>LDIF</td>
<td>LDAP Data Interchange Format. The format used to represent directory server entries in text form.</td>
</tr>
<tr>
<td>LEAP</td>
<td>Lightweight Extensible Authentication Protocol.</td>
</tr>
</tbody>
</table>
| MAC   | (1) Message Authentication Code. A MAC function takes a variable-length input and a key to produce a fixed-length output to carry authentication and integrity protection of data.  
(2) Media Access Control. The unique hardware address associated with a computer network interface. |
<p>| managed device | A device that runs an SNMP agent. |</p>
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>management station</td>
<td>Host that monitors and controls managed devices running SNMP agents.</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base. A database of objects, such as alarm status or statistics counters, that can be monitored or overwritten by an SNMP management station.</td>
</tr>
<tr>
<td>MS-CHAP</td>
<td>Microsoft CHAP. Proprietary version of CHAP.</td>
</tr>
<tr>
<td>NAS</td>
<td>Network Access Server. Network device that accepts connection requests from remote users, authenticates users via RADIUS, and routes user onto the network. Identical in meaning to RAS.</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation. Technique that allows an intranet to use IP addresses that are different from what the Internet recognizes.</td>
</tr>
<tr>
<td>native user</td>
<td>A user authenticated by Steel-Belted Radius using its internal authentication database.</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity. Standard (open) application programming interface for accessing a database.</td>
</tr>
<tr>
<td>OTP token</td>
<td>One-time password token. Hardware or software module that generates one-time passwords that can be used to authenticate a user.</td>
</tr>
<tr>
<td>PAC</td>
<td>Protected Access Credential. A high-entropy secret that is known to both the RADIUS client and the RADIUS server to secure the TLS handshake in EAP-FAST authentication.</td>
</tr>
<tr>
<td>PAP</td>
<td>Password Authentication Protocol. An authentication protocol where a requestor sends an identifier and password to a server after a link has been established. If the identifier and password match an entry in the server’s database, the authentication is acknowledged.</td>
</tr>
<tr>
<td>PEAP</td>
<td>Protected Extensible Authentication Protocol. A two-phase authentication protocol where (1) an authentication server is authenticated to a supplicant using a digital certificate and a secure channel is established; and (2) the supplicant is authenticated to the authentication server via the secure channel.</td>
</tr>
<tr>
<td>POTP</td>
<td>Protected One-Time Password. EAP method that uses one-time password tokens for unilateral or mutual authentication.</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-Point Protocol. Network protocol defined in RFC 1661 that provides a standard method for transporting multi-protocol datagrams over point-to-point links.</td>
</tr>
<tr>
<td>proxy RADIUS</td>
<td>Process of authenticating users whose profiles are on other RADIUS servers by forwarding access-request packets received from a RADIUS client to a remote RADIUS server (the proxy target), and then forwarding the response from the remote server back to the RADIUS client.</td>
</tr>
<tr>
<td>proxy target</td>
<td>The remote RADIUS server that actually performs authentication in a proxy RADIUS sequence.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Authentication Dial In User Service. A client/server security administration standard that functions as an information clearinghouse, storing authentication information about users and administering multiple security systems across complex networks.</td>
</tr>
<tr>
<td>RAS</td>
<td>Remote Access Server. Network device that accepts connection requests from remote users, authenticates users via RADIUS, and routes user onto the network. Identical in meaning to NAS.</td>
</tr>
<tr>
<td>return list</td>
<td>A list of attributes that Steel-Belted Radius must return to a RADIUS client after authentication of a user succeeds. The return list usually provides additional parameters that the RADIUS client needs to complete the connection.</td>
</tr>
<tr>
<td>roaming</td>
<td>The ability to move from one Access Point coverage area to another without interruption of service or loss of connectivity.</td>
</tr>
<tr>
<td>RSA SecurID</td>
<td>Security token system that allows remote-access users to generate a pseudorandom value they can forward as part of an authentication sequence.</td>
</tr>
<tr>
<td>session ID</td>
<td>Session Identifier. A string of characters uniquely identifying the session.</td>
</tr>
<tr>
<td>SHA-1</td>
<td>Secure Hash Algorithm-1. A one-way cryptographic function that takes a message of any length and produces a 160-bit message digest.</td>
</tr>
<tr>
<td>shared secret</td>
<td>An encryption key known only to the sender and receiver of data.</td>
</tr>
<tr>
<td>silent discard</td>
<td>The process of discarding a packet without further processing and without notification to the sender.</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer. Program layer that manages the security of messages on a network.</td>
</tr>
<tr>
<td>supplicant</td>
<td>The client in an 802.1X-authenticated network.</td>
</tr>
<tr>
<td>TACACS+</td>
<td>Terminal Access Controller Access Control System (with enhancements). An authentication protocol that allows a RAS to communicate with an authentication server to determine if a user should have access to a protected network.</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security.</td>
</tr>
<tr>
<td>trap</td>
<td>An SNMP message that reports a significant event, such as a problem, error, or change in state, that occurred within a managed device.</td>
</tr>
<tr>
<td>trap sink</td>
<td>The destination for trap messages sent by an SNMP agent on a managed device.</td>
</tr>
<tr>
<td>TTLS</td>
<td>Tunneled Transport Layer Security.</td>
</tr>
<tr>
<td>user database</td>
<td>A database where a RADIUS server keeps information about users, such as authentication information and network access permissions.</td>
</tr>
<tr>
<td>user profile</td>
<td>A record in the user database that describes how a particular user or class of users should be configured during authentication and authorization.</td>
</tr>
<tr>
<td>Abbr.</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>VSA</td>
<td>Vendor Specific Attributes.</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy. An encryption method designed to encrypt traffic between a WLAN client and an access point.</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network.</td>
</tr>
</tbody>
</table>
Appendix B

When to Restart Steel-Belted Radius

This appendix explains the following topics when to stop and restart Steel-Belted Radius.

The least drastic action that causes this change to take effect is indicated by Yes in this table:

Table 41: When to Stop and Restart Steel-Belted Radius

<table>
<thead>
<tr>
<th>Item changes:</th>
<th>Save the window or file</th>
<th>Stop/restart the server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access window or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>*.acc files</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>account.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>*.aut files</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Authentication policy</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>*.dct files</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>*.eap files</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>eap.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>events.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Import *.rif or users file</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>IP Pools dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>IPX Pools dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Log levels (in radius.ini file)</td>
<td>No</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Profiles dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Proxy dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>radius.dct file (see Notes below)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>radius.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RADIUS Clients dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Servers dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>services file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>tacplus.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TLS and TTLS</td>
<td>No</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Trace levels</td>
<td>No</td>
<td>(Also works)</td>
</tr>
<tr>
<td>Tunnels panel or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
</tbody>
</table>
Table 41: When to Stop and Restart Steel-Belted Radius (continued)

<table>
<thead>
<tr>
<th>Item changes:</th>
<th>Save the window or file</th>
<th>Stop/restart the server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users dialog or object</td>
<td>Yes</td>
<td>(Also works)</td>
</tr>
<tr>
<td>vendor.ini file</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
This appendix contains the following technical bulletins:

- LDAP Support for Novell eDirectory
- CCA Support for 3COM
- Ascend Filter Translation
- Uniport Plug-In
- Windows Performance Monitor

### LDAP Support for Novell eDirectory

The Steel-Belted Radius LDAP authentication plug-in contains features to enable greater interoperability with Novell eDirectory.

If you have configured eDirectory to limit the number of grace logins available to a user, Steel-Belted Radius can be configured to coordinate with eDirectory on this feature. Each time a user authenticates, the number of grace logins available is decremented until the account is locked out and needs an administrator to unlock it. A profile is assigned to these users — a profile that overrides their normal profile — when they are being authenticated using a grace login.

The features include:

- **Allowing Expired Accounts**: When enabled, this feature allows users to log in even after their account has expired.

**NOTE:** As eDirectory itself does not check the password, this feature should be used only if the administrator has configured the ProfileForExpiredUsers setting to assign an alternate profile to the user, one that would inform the user of their account status but not allow a usable connection to the network. For example, you can use http redirection to force the connection to a web page with relevant information.
NOTE: Administrators should contact their NAS vendors to determine the capabilities of their NAS equipment and what attribute-value pairs would be needed to create such a connection.

NOTE: Grace Logins: When grace logins are limited in eDirectory, Steel-Belted Radius can be configured to accept or reject a user whose password has expired. This user is said to be in grace login mode; the user can also be allowed to log in but is provided with an alternate profile.

The grace login feature requires NetWare 6.0 or later with eDirectory 8.6 or later.

- **BindName**: You can use the BindName technique to search the eDirectory directory for a matching user and, having retrieved the user’s DN, apply the Bind technique to authenticate the user’s credentials. This combination allows the user to specify their common name (rather than the more cumbersome DN) when requesting authentication.

NOTE: This feature works only if the NetWare server accepts LDAP Bind requests for users who are in grace login state. Earlier versions of the NetWare server did not support this feature.

NOTE: You must configure the eDirectory server to ‘Allow Clear_text passwords’ to use these LDAP extensions. You can do this from the ConsoleOne application, in the Properties section of the LDAP group entry for your server.

### Configuration

The [NDS] section (Table 42) has been added to the `ldapauth.aut` file to configure these features.

#### Table 42: [NDS] Settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Set to 1 to enable the Novell eDirectory (NDS) extensions. Default value is 0.</td>
</tr>
</tbody>
</table>
| AllowExpiredAccountsForUsers | Set to 1 to allow users to authenticate even after their account has expired. Default value is 0.  
**Important**: This requires that the Netware server notify Steel-Belted Radius that the user’s account has expired when the Netware server is attempting to Bind. |
| ProfileForExpiredUsers  | The name of the profile to assign a user (as an override) if authenticated with an expired account. If you do not provide a value for this setting, the user is accepted with the usual profile and attributes. We recommend that you provide the user with a profile with restricted access. |
If Enable is set to 1, Steel-Belted Radius works as follows:

- If the eDirectory directory is configured to operate without grace logins:
  - If `AllowExpiredAccountsForUsers` is set to 0 (the default), users with expired passwords are rejected.
  - If `AllowExpiredAccountsForUsers` is set to 1, users with expired accounts are accepted; the attributes returned in the Access-Accept response are either the attributes normally assigned to the user or, if the `ProfileForExpiredUsers` setting is specified, the attributes specified in that profile. If you enable this feature, we recommend that you configure the `ProfileForExpiredUsers` setting.

- If the eDirectory directory is configured to operate with grace logins:
  - If `AllowGraceLoginsForUsers` is set to 0, users with correct but about-to-expire passwords are rejected.
  - If `AllowGraceLoginsForUsers` is set to 1 (the default), users with correct but about-to-expire passwords are accepted; the attributes returned in the Access-Accept are either the regular attributes assigned to the user or, if `ProfileForGraceLoginUsers` is specified, the attributes specified in that profile.

### Sample ldapauth.aut file

```
[Bootstrap]
LibraryName=ldapauth.dll
Enable=1
InitializationString=LDAP

[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
BindName=uid=<User-Name>, ou=sales, o=bigco.com
LogLevel = 0
UpperCaseName = 0
PasswordCase=original
PasswordFormat=0
```
Search = DoLdapSearch
SSL = 0

[Server]
s1=

[Server/s1]
Host=192.168.5.110
Port = 389

[Request]
%UserName = User-Name

[Response]
%profile= attThatContainsUserProfile

[Search/DoLdapSearch]
;Bind as a privileged user or someone that has the right to
; search the tree and retrieve the DN of users
bind=cn=administrator,o=netware6
Password= support
Base = o=netware6
Scope = 2
Filter = uid=<User-Name>
%DN = dn
;if the user is found perform search getprofile
onfound = GetProfile
;else reject the user
onnfound=$reject

[Search/GetProfile]
; bind using the DN retrieved in doLdapSearch
Bind = <dn>
;You do not have to supply the password SBR knows to use the
;one received in the auth request.
;Setting base to the DN is most efficient
Base =<DN>
Scope = 2
Filter = uid=<User-Name>
Attributes = AttrList

[Attributes/AttrList]
attThatContainsUserProfile

[NDS]
Enable=1
AllowExpiredAccountsForUsers=1
ProfileForExpiredUsers=Expired
AllowGraceLoginsForUsers=1
ProfileForGraceLoginUsers=Grace

[Attributes/AttrList]
;Filter-Id
;Session-Timeout
;thepasswordis
CCA Support for 3COM

Steel-Belted Radius can support the generation of 3Com CCA tunnel attributes.

Configuration

To enable the return of the required CCA tunnel attributes, the ccagw.ini file must be modified.

The ccagw.ini file contains information about gateways, which are stored in the [gateway] sections. A [gateway] section must be present for each gateway supported.

Table 43 describes each field:

Table 43: ccagw.ini File

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address of the gateway.</td>
</tr>
<tr>
<td>TunnelRefresh</td>
<td>The number of seconds before the tunnel refreshes. The default value is 0.</td>
</tr>
<tr>
<td>Description</td>
<td>A text string describing the gateway.</td>
</tr>
<tr>
<td>Secret</td>
<td>The shared secret between Steel-Belted Radius and this gateway device.</td>
</tr>
</tbody>
</table>

For example:

[Jupiter-Gateway]
Address = 200.47.98.142
TunnelRefresh = 3600
Description = Jersey City facility, East Coast subscribers
Secret = Holland Tunnel

Setting User and Profile Attributes

To enable this functionality for a particular user, the return list for the user must contain the following attributes:

Tunnel-Authentication
VPN-Gateway

Both of these attributes are defined as strings. The value of each attribute must be set to the name of the gateway used in the ccagw.ini file. For example, the return list of a user would have to include:

Tunnel-Authentication = Jupiter-Gateway
VPN-Gateway = Jupiter-Gateway

It is important to make sure that both attributes name the correct gateway. If an unknown gateway is named, the request is rejected.

NOTE: Steel-Belted Radius is capable of returning multiple pairs of attributes for different gateways. For each gateway named in one of the attributes, a different random session key is generated.
Ascend Filter Translation

Ascend defines two attributes — Ascend-Data-Filter (242) and Ascend-Call-Filter (243) — that contain structured binary data representing a filter to be applied to the NAS device.

Instead of entering hexadecimal strings to configure these attributes, users can configure these attributes as text strings. Steel-Belted Radius automatically converts the text strings to the proper binary representation. The original filter attributes are still supported, and these attributes still may be configured as hexadecimal strings.

The following attributes allow configuration as text:

- Ascend-Data-Filter-String
- Ascend-Call-Filter-String

When Steel-Belted Radius formats a response packet, it translates the string version of the attribute to the appropriate binary value, and returns the attribute in the Access-Accept message.

Configuration

These attributes may be entered as text strings through the SBR Administrator. The attributes may also be returned from an LDAP or SQL database during authentication.

No syntax validation is performed when the attribute is configured. The validation of syntax occurs only when the response packet is formatted. If the syntax is invalid, a reject response is issued and an error is logged.

NOTE: These attributes should be tested before configured on a production server.

Two types of filter are supported: “ip” and “generic”. “ipx” filters are not supported.

Syntax

In the syntax descriptions in Table 44, brackets [ ] indicate that the items enclosed are optional.

\[
ip [\text{direction}] [\text{action}] [\text{srcip address}/\text{mask}] [\text{dstip address}/\text{mask}] \text{ protocol [srcport operator port]} [\text{dstport operator port}]
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>direction</td>
<td>May be &quot;in&quot; or &quot;out&quot;. The default is &quot;out&quot;.</td>
</tr>
<tr>
<td>action</td>
<td>May be &quot;forward&quot; or &quot;drop&quot;. The default is &quot;drop&quot;.</td>
</tr>
</tbody>
</table>
Table 44: Ascend Filter Translation parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>An IP address in decimal dotted notation.</td>
</tr>
<tr>
<td>mask</td>
<td>The number of bits (decimal) in the network portion, from 0 through 32. The default is based on class of network.</td>
</tr>
<tr>
<td>protocol</td>
<td>The protocol number (decimal); for example, 6 for TCP or 17 for UDP. The following protocol names are translated to the proper number: tcp(6), udp(17), ospf(89).</td>
</tr>
<tr>
<td>operator</td>
<td>May be = (equal sign), != (exclamation and equal sign), &lt; (less than), or &gt; (greater than).</td>
</tr>
<tr>
<td>port</td>
<td>The port number (decimal). In addition, the following service names are translated to the proper port number: ftp-data(20), www(80), ftp(21), kerberos(88), telnet(23), hostname(101), smtp(25), nntp(119), nameserver(42), ntp(123), domain(53), exec(512), tftp(69), login(513), gopher(70), cmd(514), finger(79), talk(517).</td>
</tr>
</tbody>
</table>

Example:

```
ip out forward srcip 10.1.1.0/24 dstport = 80 srcport < 1023
```

NOTE: Please see your Ascend documentation for details about the syntax for these attributes.

# Ericsson Enhanced Token Caching

This section pertains only to Ericsson equipment that supports enhanced token caching.

Enhanced token caching allows the administrator to specify that particular users are authenticated with both an ordinary password and a SecurID passcode. For such users, the ordinary PAP or CHAP password is checked first. If this first
authentication is successful, the user’s SecurID passcode is authenticated. Only if both authentications succeed is the user allowed access.

---

**NOTE:** The enhanced token caching feature does not interact with the ordinary token caching feature described in this section. Enhanced token caching is required to support the newer firmware releases on Ericsson devices.

### Enhanced Token Caching Configuration

To enable enhanced token caching, a file with extension `.aut` must be included in the server directory—typically named `sidalt.aut`. The file has the following format:

```plaintext
[Bootstrap]
LibraryName = sidalt.dll
Enable = \{ 0 | 1 \}
InitializationString = string

[Settings]
TokenAttr = string
CacheTimeoutAttr = string
MessageID = \{ 0 | 1 \}
ChallengeTokenInPassword = \{ 0 | 1 \}
```

Table 45 describes each field in the `sidalt.aut` file.

<table>
<thead>
<tr>
<th>Sidalt.aut Field</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| Enable                   | - Set to 0 to disable.  
- Set to 1 to enable.  
Default value is 0. |
| LibraryName              | Identifies the dynamic link library used to process SecurID authentication.  
Default value is `sidalt.dll`. |
| InitializationString     | Identifies the enhanced token caching software component for logging purposes. A typical value might be `SecurID Alt`.  
This field is required. |
| TokenAttr                | The name of the Access-Request attribute containing the passcode or other information to passed to the RSA SecurID server. This attribute must match the corresponding dictionary (.dct file) entry.  
This field is required. |
| CacheTimeoutAttr         | The name of the Access-Response attribute containing the number of seconds a passcode remains in the cache from the time it was first validated by the RSA SecurID server. This attribute must match the corresponding dictionary (.dct file) entry.  
This field is required. |
Enhanced Token Caching Administration

To authenticate a user through the enhanced token caching component, the following must be true:

- The attribute specified by the TokenAttr entry must be present in the Access-Request.
- The return list specified for the user, either directly or through a profile, must include the attribute specified by the CacheTimeoutAttr entry.

If either attribute is not supplied, the user is assumed not to require enhanced token caching authentication, and is accepted.

NOTE: See your Ericsson product documentation for information about NAS and PPP client operation under the enhanced token caching authentication method.

Unisport Plug-In

3Com’s Uniport project can operate with Steel-Belted Radius via a plug-in.

Operation

Uniport requires RADIUS call-type determination as a back-up for SIP call-type determination. To determine call-type, the HiPerARC system sends Steel-Belted Radius a request containing a Service-Type attribute of Call-Check (10) and a User-Name attribute in which the value is the same as the Called-Station-Id (DNIS) attribute. The type of call is then determined based on the User-Name (DNIS), and the appropriate Service-Type attribute returned in the Access-Accept packet.
A Uniport plug-in method is instantiated for each value of the Service-Type attribute which can be returned in the Access-Accept. The proper method is utilized using proxy mapping to a directed realm which specifies the method instance. The method then sets the configured profile in the response and indicates it was successful.

The Uniport methods return a Reject if a Service-Type attribute with a Call-Check value is not present in the request, if User-Name or Called-Station-Id attributes are not present, or if their values are not identical.

**Configuration**

The attribute(s) to be returned in the Access-Accept to identify the call-type are defined as Steel-Belted Radius profiles. For example, a FAX profile may be created which would return a value of 96 in a Service-Type attribute.

The Uniport methods are configured with *.AUT files which specify the profile to return. The method is identified and associated with a directed realm by the initialization string.

For example, a method to return the FAX profile may use a configuration file such as `FAX.AUT`, which would have the following settings:

```yaml
[Bootstrap]
Enable = 1
InitializationString = UNIPORT FAX
Profile = FAX
LibraryName = Uniport.so
```

The corresponding directed realm would then identify the method in its *.DIR file. For example, in the `FAX.DIR` file the settings might be:

```yaml
[Auth]
Enable = 1

[AuthMethods]
Uniport fax
```

If you want a default profile, the map may be configured to direct the request to a Uniport method by default. The same result may be obtained by omitting the default from the map and setting the first method in the authentication method chain to the desired default.

If you do not want a default profile, configure the map to direct requests by default to a method which has no profile set; the method returns with a value to indicate a failure to authenticate.
Windows Performance Monitor

The Steel-Belted Radius service has information which can be viewed with the Performance Monitor on a Windows administrative workstation or server. You can start multiple instances of the Windows Performance Monitor to display performance statistics graphs for more than one Steel-Belted Radius server simultaneously.

To view a graph of Steel-Belted Radius performance:


2. Click the + button in the Performance window (or press CTRL + I).

   The Add Counters window opens.

3. Pull down the Performance Object list and choose Steel-Belted Radius.

   If you are running multiple Steel-Belted Radius servers, select the one you want to monitor from the Select counters from computer list.

4. If you want to select the counters that you want to graph, click the Select counters from list radio button, choose the counter you want, and click Add. Continue choosing counters and clicking Add until all desired counters have been selected.

   If you want to display all counters for Steel-Belted Radius, click the All counters radio button and click Add.

   Most perfmon counters relating to Steel-Belted Radius have self-explanatory names, such as Acct Failures - Insufficient Resources or Acct Failures - Invalid Shared Secret.

   Of special interest is the Failed Auths - n counter, where n is a number between 1 and 16. You can set up as many as 16 Failed Auths - n counters, where each counter tracks the number of failed authentication requests that were encountered for all of the RADIUS clients that you have mapped to collection number n.

   To set up the Failed Auths - n counter, you must configure the [FailedAuthOriginStats] section of radius.ini. For information on radius.ini, refer to the Steel-Belted Radius Reference Guide.

5. Click Close when you are finished adding counters.

   The Performance Monitor window displays a graph of the counters you have selected. The graph updates itself at regular intervals until you close the Performance Monitor window.

6. Optionally, specify the color, scale, line width, and line style for one or more Steel-Belted Radius counters.
Right-click the name of a counter and choose **Properties** from the context menu. When the System Monitor Properties window opens, specify the characteristics you want the graph of the counter to use.

Table 46 describes the meaning of each perfmon counter.

### Table 46: perfmon Counters

<table>
<thead>
<tr>
<th>perfmon Counter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acct Failures - Insufficient Resources</td>
<td>The number of accounting requests that were discarded because the RADIUS server was unable to obtain sufficient system resources to process the request.</td>
</tr>
<tr>
<td>Acct Failures - Invalid Clients</td>
<td>The number of accounting requests that were discarded because the RADIUS client identified in the request was not defined in the RADIUS server database.</td>
</tr>
<tr>
<td>Acct Failures - Invalid Requests</td>
<td>The number of accounting requests that were discarded because the request was malformed or contained invalid attributes.</td>
</tr>
<tr>
<td>Acct Failures - Invalid Shared Secret</td>
<td>The number of accounting requests that were discarded because the request contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the RADIUS client and the RADIUS server.</td>
</tr>
<tr>
<td>Acct Proxy Failures</td>
<td>The number of forwarded accounting requests for which failures were encountered.</td>
</tr>
<tr>
<td>Acct Requests Forwarded</td>
<td>The number of accounting requests that were forwarded to other RADIUS servers.</td>
</tr>
<tr>
<td>Acct Requests Retried</td>
<td>The number of unique accounting requests for which retries were received by the RADIUS server.</td>
</tr>
<tr>
<td>Acct Requests Retried/sec</td>
<td>The number of accounting requests per second for which one or more retries has been received by the RADIUS server.</td>
</tr>
<tr>
<td>Acct Retry Requests</td>
<td>The number of actual accounting request retries received by the RADIUS server.</td>
</tr>
<tr>
<td>Acct Retry Requests/sec</td>
<td>The number of accounting request retries per second received by the RADIUS server.</td>
</tr>
<tr>
<td>Acct Service Time</td>
<td>The number of seconds that elapsed from the time the last completed accounting request was received to the time the RADIUS server sent a response. Responses generated by proxies are not reflected in this statistic.</td>
</tr>
<tr>
<td>Acct Starts</td>
<td>The number of accounting start requests received by the RADIUS server. An accounting start signifies the granting of a connection to an end-user by the remote access server.</td>
</tr>
<tr>
<td>Acct Starts/sec</td>
<td>The number of accounting start requests received by the RADIUS server per second. An accounting start signifies the granting of a connection to an end-user by the remote access server.</td>
</tr>
<tr>
<td>Acct Stops</td>
<td>The number of accounting stop requests received by the RADIUS server. An accounting stop signifies that an end-user has disconnected from the remote access server.</td>
</tr>
<tr>
<td>Acct Stops/sec</td>
<td>The number of accounting stop requests received by the RADIUS server per second. An accounting stop signifies that an end-user has disconnected from the remote access server.</td>
</tr>
<tr>
<td>Auth Failure - Authentication Failures</td>
<td>Number of unique authentication requests to which the RADIUS server replied with a reject because no user specified in the database possessed a matching password. A mismatch in shared secrets would also cause this counter to be incremented.</td>
</tr>
</tbody>
</table>
Table 46: perfmon Counters (continued)

<table>
<thead>
<tr>
<th>perfmon Counter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth Failure - Checklist Mismatches</td>
<td>Number of unique authentication requests to which the RADIUS server replied with a reject because the request did not include required checklist information.</td>
</tr>
<tr>
<td>Auth Failure - Insufficient Resources</td>
<td>Number of unique authentication requests to which the RADIUS server replied with a reject because the RADIUS server ran into a system resource limitation.</td>
</tr>
<tr>
<td>Auth Failure - Invalid Clients</td>
<td>Number of unique authentication requests to which the RADIUS server replied with a reject because the request was from a RADIUS client not identified in the RADIUS server’s database.</td>
</tr>
<tr>
<td>Auth Failure - Invalid Requests</td>
<td>Number of unique authentication requests to which the RADIUS server replied with a reject because the request was malformed or contained invalid attributes.</td>
</tr>
<tr>
<td>Auth Proxy Failures</td>
<td>The number of forwarded authentication requests for which failures were encountered.</td>
</tr>
<tr>
<td>Auth Proxy Rejects</td>
<td>The number of forwarded authentication requests for which rejects were received from the target RADIUS server.</td>
</tr>
<tr>
<td>Auth Requests</td>
<td>The number of unique authentication requests that the RADIUS server has received.</td>
</tr>
<tr>
<td>Auth Requests Forwarded</td>
<td>The number of authentication requests that were forwarded to another RADIUS server.</td>
</tr>
<tr>
<td>Auth Requests Retried</td>
<td>The number of unique authentication requests for which retries were received by the RADIUS server.</td>
</tr>
<tr>
<td>Auth Requests Retried/sec</td>
<td>The number of authentication requests per second for which one or more retries has been received by the RADIUS server.</td>
</tr>
<tr>
<td>Auth Requests/sec</td>
<td>The number of unique authentication requests that the RADIUS server has received per second.</td>
</tr>
<tr>
<td>Auth Retry Requests</td>
<td>The number of actual authentication request retries received by the RADIUS server.</td>
</tr>
<tr>
<td>Auth Retry Requests/sec</td>
<td>The number of authentication request retries per second received by the RADIUS server.</td>
</tr>
<tr>
<td>Auth Service Time</td>
<td>The number of seconds that elapsed from the time the last completed authentication request was received to the time the RADIUS server sent an Accept response. Accept responses generated for tunnel requests or by proxies are not reflected in this statistic.</td>
</tr>
<tr>
<td>Auth SQL Disconnects</td>
<td>The number of times an existing connection to a SQL authentication database failed.</td>
</tr>
<tr>
<td>Auth SQL Failures</td>
<td>The number of times an attempt to connect to a SQL authentication database failed.</td>
</tr>
<tr>
<td>Auth SQL Records Not Found</td>
<td>The number of times no record was found in a SQL authentication database for the specified username.</td>
</tr>
<tr>
<td>Auth SQL Timeouts</td>
<td>The number of times a timeout occurred attempting to execute a SQL authentication request.</td>
</tr>
<tr>
<td>Auth Successes</td>
<td>The number of unique authentication requests to which the RADIUS server replied with an Accept.</td>
</tr>
<tr>
<td>Auth Successes/sec</td>
<td>The number of unique authentication requests to which the RADIUS server replied with an accept per second.</td>
</tr>
<tr>
<td>perfmon Counter</td>
<td>Meaning</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Concurrency Auth Failures</td>
<td>The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject for reasons other than users being over their port limits.</td>
</tr>
<tr>
<td>Concurrency Auth Service Time</td>
<td>The number of seconds elapsed from the last time an authentication request was sent to the concurrency server and a response was received.</td>
</tr>
<tr>
<td>Concurrency Auth Timeouts</td>
<td>The number of times an authentication request was forwarded to the concurrency server and no response was received within the configured time for the proxy entry.</td>
</tr>
<tr>
<td>Concurrency Over Port Limit</td>
<td>The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject because users were over their port limits.</td>
</tr>
<tr>
<td>Failed Auths - 1</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 1. To set up the Failed Auths - n counter, you must configure the [FailedAuthOriginStats] section of radius.ini. For information on radius.ini, refer to the Steel-Belted Radius Reference Guide.</td>
</tr>
<tr>
<td>Failed Auths - 2</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 2.</td>
</tr>
<tr>
<td>Failed Auths - 3</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 3.</td>
</tr>
<tr>
<td>Failed Auths - 4</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 4.</td>
</tr>
<tr>
<td>Failed Auths - 5</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 5.</td>
</tr>
<tr>
<td>Failed Auths - 6</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 6.</td>
</tr>
<tr>
<td>Failed Auths - 7</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 7.</td>
</tr>
<tr>
<td>Failed Auths - 8</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 8.</td>
</tr>
<tr>
<td>Failed Auths - 9</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 9.</td>
</tr>
<tr>
<td>Failed Auths - 10</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 10.</td>
</tr>
<tr>
<td>Failed Auths - 11</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 11.</td>
</tr>
<tr>
<td>Failed Auths - 12</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 12.</td>
</tr>
<tr>
<td>Failed Auths - 13</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 13.</td>
</tr>
<tr>
<td>Failed Auths - 14</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 14.</td>
</tr>
<tr>
<td>Failed Auths - 15</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 15.</td>
</tr>
<tr>
<td>Failed Auths - 16</td>
<td>The number of failed authentication requests that were encountered for clients categorized in collection number 16.</td>
</tr>
</tbody>
</table>
### Table 46: perfmon Counters (continued)

<table>
<thead>
<tr>
<th>perfmon Counter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarded Requests Retried</td>
<td>The number of unique forwarded accounting and authentication requests for which retries were transmitted by the RADIUS server.</td>
</tr>
<tr>
<td>Forwarded Requests Retried/sec</td>
<td>The number of unique forwarded accounting and authentication requests for which retries were transmitted per second by the RADIUS server.</td>
</tr>
<tr>
<td>Forwarded Retry Requests</td>
<td>The number of actual retransmissions of forwarded accounting and authentication request performed by the RADIUS server.</td>
</tr>
<tr>
<td>Forwarded Retry Requests/sec</td>
<td>The number of actual retransmissions of forwarded accounting and authentication request per second performed by the RADIUS server.</td>
</tr>
<tr>
<td>Proxy Failures - Insufficient Resources</td>
<td>The number of authentication and accounting requests that were forwarded to other RADIUS servers for which the RADIUS server was unable to obtain sufficient system resources to process the request.</td>
</tr>
<tr>
<td>Proxy Failures - Invalid Response</td>
<td>The number of authentication and accounting requests that were forwarded to other RADIUS servers for which malformed or invalid responses were received.</td>
</tr>
<tr>
<td>Proxy Failures - Invalid Shared Secret</td>
<td>The number of authentication and accounting requests that were forwarded to other RADIUS servers for which responses were discarded because the response contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the RADIUS client and RADIUS server.</td>
</tr>
<tr>
<td>Proxy Failures - Time Out</td>
<td>The number of authentication and accounting requests that were forwarded to other RADIUS servers for which no response was received after the specified number of retries.</td>
</tr>
<tr>
<td>Seconds since started</td>
<td>Number of seconds Steel-Belted Radius has been running.</td>
</tr>
<tr>
<td>Sessions Online</td>
<td>The number of sessions currently active in the RADIUS server’s Sessions list.</td>
</tr>
<tr>
<td>Static Acct Service Time</td>
<td>The number of seconds elapsed from the last time an accounting request was sent to the static accounting proxy server and a response was received.</td>
</tr>
<tr>
<td>Total Acct Failures</td>
<td>The number of unique accounting requests to which the RADIUS server did not reply. Reasons for the failures are identified in other statistics.</td>
</tr>
<tr>
<td>Total Acct Failures/sec</td>
<td>The number of unique accounting requests to which the RADIUS server did not reply per second because of an error. Reasons for the failures are identified in other statistics.</td>
</tr>
<tr>
<td>Total Acct Offs</td>
<td>The number of accounting off requests received by the RADIUS server. An accounting off signifies that the accounting support in the RADIUS client has been disabled. This request is most often issued when a RADIUS client in being shut down.</td>
</tr>
<tr>
<td>Total Acct Offs/sec</td>
<td>The number of accounting off requests received by the RADIUS server per second.</td>
</tr>
<tr>
<td>Total Acct Ons</td>
<td>The number of accounting on requests received by the RADIUS server. An accounting on signifies that the accounting support in the RADIUS client has been enabled. This request is most often issued when a RADIUS client is powered on.</td>
</tr>
<tr>
<td>Total Acct Ons/sec</td>
<td>The number of accounting on requests received by the RADIUS server per second.</td>
</tr>
<tr>
<td>Total Auth Challenges</td>
<td>The number of authentication requests that resulted in a RADIUS challenge response.</td>
</tr>
</tbody>
</table>
### Table 46: perfmon Counters (continued)

<table>
<thead>
<tr>
<th>perfmon Counter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Auth Failures</td>
<td>The number of unique authentication requests to which the RADIUS server replied with a reject. Reasons for the failures are identified in other statistics.</td>
</tr>
<tr>
<td>Total Auth Failures/sec</td>
<td>The number of unique authentication requests to which the RADIUS server replied with a reject, per second. Reasons for the failures are identified in other statistics.</td>
</tr>
<tr>
<td>Total Forwarded Request Failures</td>
<td>The number of forwarded authentication and accounting requests that encountered failures.</td>
</tr>
<tr>
<td>Total Forwarded Request Failures/sec</td>
<td>The number of forwarded authentication and accounting requests that encountered failures, per second.</td>
</tr>
<tr>
<td>Total Forwarded Requests</td>
<td>The number of authentication and accounting requests that were forwarded to other RADIUS servers.</td>
</tr>
<tr>
<td>Total Forwarded Requests/sec</td>
<td>The number of authentication and accounting requests per second that were forwarded to other RADIUS servers.</td>
</tr>
<tr>
<td>Users Online</td>
<td>The number of unique users represented in the RADIUS server’s Sessions List.</td>
</tr>
</tbody>
</table>
Appendix D

Authentication Protocols

This appendix provides a matrix of authentication methods and their supported authentication protocols.

Table 47: Authentication Protocols

<table>
<thead>
<tr>
<th>Method</th>
<th>PAP</th>
<th>CHAP</th>
<th>MS-CHAP</th>
<th>MS-CHAP-V2</th>
<th>LEAP</th>
<th>EAP-MSCHAP-V2</th>
<th>EAP-MD5</th>
<th>PAP/Token card</th>
<th>EAP/Token card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft PEAP available inner authentication protocols</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cisco PEAP available inner authentication protocols</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TTLS available inner authentication protocols</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Local</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Local (password saved as Allow PAP only; {SHA} or {crypt}).</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Windows Domain Authentication</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>Windows Domain Group</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Windows Domain User</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>UNIX authentication methods</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>UNIX User</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>UNIX Group</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Other authentication plug-ins</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA SecurID</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>TACACS +</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>LDAP</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BIND (this includes AD and eDirectory/NDS)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BINDNAME (password stored in clear text)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BINDNAME (password stored in SHA/Solaris Crypt text)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BINDNAME (password stored as MD4 hash of unicode value text)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 47: Authentication Protocols (continued)

<table>
<thead>
<tr>
<th>Method</th>
<th>PAP</th>
<th>CHAP</th>
<th>MS-CHAP</th>
<th>MS-CHAP-V2</th>
<th>LEAP</th>
<th>EAP-MSCHAP-V2</th>
<th>EAP-MD5</th>
<th>PAP/Token card</th>
<th>EAP/Token card</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINDNAME (password stored as enc-md5)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SQL</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Password stored in clear text</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Password password stored in SHA/Solaris Crypt text</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Password stored as {MD4} hash of unicode value text</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Password stored as {enc-md5}</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Appendix E

Importing and Exporting Data

This appendix describes how to export database information from one Steel-Belted Radius server to an XML file, and then selectively import database information into another Steel-Belted Radius server. The ability to export and import information facilitates configuration of multiple Steel-Belted Radius servers.

Steel-Belted Radius uses XML files with UTF-8 encoding to store exported information. Export files contain only those attributes that have been assigned values, including defaults. For example, if an exported local user item does not have a profile associated with it, the exported information will not identify a profile name, though it will include default values required by Steel-Belted Radius.

NOTE: Versions of Steel-Belted Radius earlier than Version 5.0 used a file format called RADIUS Import File (RIF) to export and import information. For backward compatibility, Steel-Belted Radius includes a utility to convert RIF files to XML format. For more information, see the Steel-Belted Radius Getting Started manual.

Exporting to a RADIUS Information File

To export information from the Steel-Belted Radius database to an XML file:

1. Run the SBR Administrator.
2. Choose File > Export.
3. When the Export dialog (Figure 129) opens, select the information you want to export.

   Each tab in the dialog lists exportable items of a particular category. For each category, select the appropriate tab and click each item you'd like to export.

   To select a contiguous range of items, select the first item in the range, hold down the Shift key, and click the last item in the range.

   To select a non-contiguous set of items, hold down the Ctrl key as you click each item you want.

   To select all items in a category, click All.

   To select all items in all categories, click Select All.
4. After you have selected the items you want to export, click **OK**.

5. When the Export to XML dialog (Figure 130) opens, specify a file name and click **Save**.

![Figure 130: Export to XML Dialog](image)
**Importing Into the Steel-Belted Radius Database**

To import information from an XML file into the database on your Steel-Belted Radius server:

1. Run the SBR Administrator.
2. Choose **File > Import**.
3. When the Import from XML dialog (Figure 131) opens, select the XML file containing the information you want to import and click **Open**.

**Figure 131: Import from XML File Dialog**

4. When the Import dialog (Figure 132) opens, specify whether what the SBR Administrator should do when it finds an object with the same name in the Steel-Belted Radius database.

- Click **Skip** if you want SBR Administrator to leave the item already in the database intact.
- Click **Replace** if you want SBR Administrator to overwrite the item in the database with the imported information.

5. Select the information you want to import by clicking each tab and selecting items to import.

   To select a contiguous range of items, select the first item in the range, hold down the **SHIFT** key, and click the last item in the range.

   To select a non-contiguous set of items, hold down the **Ctrl** key as you click each item you want.

   To select all items in a category, click **All**.

   To select all items in all categories, click **Select All**.
6. After you select all the items you want, click **OK**. The items you selected are added to the Steel-Belted Radius database.
Appendix F

Configuring IPv6 Support

This appendix presents an overview of current IPv6 functions in Steel-Belted Radius and describes how to configure Steel-Belted Radius to enable IPv6 addressing and attribute processing.

NOTE: Version 5.x of Steel-Belted Radius provides early support for the evolving IPv6 protocol suite to meet the needs of early-adopter environments, such as carrier test laboratories. IPv6 functions are intended for users who are actively experimenting with IPv6 networks. Enabling the IPv6 functionality in the Steel-Belted Radius v6.x server in a production environment is not recommended. We are committed to providing production-quality IPv6 support in future versions of Steel-Belted Radius.

About IPv6

IPv6 is the next step in the evolution of the Internet Protocol, currently implemented as Internet Protocol Version 4 (IPv4). IPv6, which has been under development for more than 10 years, provides improvements over IPv4 in addressing, configuration, and security. Although IPv6 is still an evolving standard, many operating system vendors offer production-quality IPv6 implementations for customers interested in experimenting with IPv6 networks.

IPv6 and Steel-Belted Radius

With few exceptions, Steel-Belted Radius supports IPv6 addressing wherever IPv4 addressing is supported. You can perform configuration, authentication, and accounting of RADIUS IPv6 attributes per RFC 3162. RADIUS and IPv6. The SBR Administrator (configuration application) and the LDAP configuration interface (LCI) support the configuration of RADIUS IPv6 attributes.

NOTE: Because many third-party libraries and software development kits (SDKs) do not support IPv6, the Steel-Belted Radius server must support local IPv4 socket connections.
**IPv6 Features**

Significant changes from IPv4 to IPv6 include the following:

- **Expanded routing and addressing capabilities**—IPv6 increases the IP address size from 32 bits to 128 bits. As a consequence, IPv6 supports more levels of addressing hierarchy, provides a much greater number of available addresses, and simplifies auto-configuration of addresses. As a consequence, address conservation techniques such as network address translation, are not necessary.

- **Improved multicast routing**—Multicast routing, which existed in IPv4, has been redefined and improved in IPv6. Multicast addresses now include a Scope field that limits the scope of multicasts, improving scalability. A new Anycast address type allows you to send a message to the nearest single member of a multicast group.

- **Header format simplification**—The IPv6 packet header format has been designed to be efficient. The IPv6 header has a fixed length of 40 bytes, divided into eight fields.

- **Improved support for extensions and options**—IPv6 uses extension headers, which are inserted between the IPv6 header and the transport header and packet payload, to handle special packet processing requirements. Extension headers provide a flexible means to support authentication, encryption, fragmentation, source routing, network management, and other functions. An IPv6 packet can carry any number of extension headers.

- **Improved datagram sizing and fragmentation**—The maximum transmission unit (MTU) describes the maximum size of a datagram that can be transmitted over a network without fragmentation. IPv6 increases the minimum MTU from 576 bytes to 1280 bytes, which makes IPv6 packets more efficient and reduces the need for packet fragmentation. Path MTU discovery enables source routers to determine the appropriate packet size for a route.

- **Quality-of-Service (QoS) functions**—Packets belonging to a traffic flow that requires special handling, such as real-time video service, can be labeled by the sender. Because traffic in a particular flow can be identified in the IPv6 header, support for QoS can be implemented even when the payload of a packet is encrypted.

- **Improved privacy and security**—IPv6 supports extensions for authentication and data integrity to improve security and privacy of network traffic.

**IPv6 Addressing**

IPv6 addresses are 128 bits in length, which creates a much larger address space than 32-bit IPv4 addresses. IPv6 addresses identify individual interfaces and sets of interfaces. IPv6 addressing architecture is defined in RFC 2373, *IP Version 6 Addressing Architecture*.

**Address Notation**

In full form, IPv6 addresses are written as eight 16-bit hexadecimal blocks separated by colons:
IPv6 addresses can be interpreted as having two variable-length fields: an IPv6 prefix and an IPv6 interface identifier.

- The IPv6 prefix varies from 0 to 128 bits in length and forms the routable network number portion of the IPv6 address. The trailing CIDR notation that may appear after human-readable IPv6 addresses (for example, /64) indicates the bit length of the IPv6 prefix.

- The IPv6 interface identifier consists of the non-prefix portion of the IPv6 address, if any, and identifies the host interface portion of the address, which identifies an IPv6 interface on the local network. The IPv6 interface identifier is typically generated automatically by the host as a function of a unique hardware identifier, such as an Ethernet MAC address. IPv6 hosts can automatically configure interface addresses by combining the IPv6 prefixes obtained from router advertisements with the IPv6 interface IDs that are determined locally.

For example:

IPv6 Prefix: FEC0:0000:0000:0000:0000:0000:0000:0000/64
IPv6 Interface ID:0260:08FF:FEFF:FFFF
IPv6 Address:FEC0:0000:0000:0000:0260:08FF:FEFF:FFFF

To simplify address notation, IPv6 accepts abbreviations in address notation. For example, leading zeros in a 16-bit block may be omitted:

FE80:0:0:0:0232:E4BF:FE1A:8324

A double colon (::) can replace a series of consecutive zeros within an address:

FE80::232:E4BF:FE1A:8324

Only one double colon can be used to compress an IPv6 address. If more than one double colon was included in an address, networking devices would not know how many zeros to insert for each double colon when expanding a compressed address to its full 128-bit representation.

In networks that support IPv4 and IPv6 nodes, IPv4 addresses can be embedded in the last four bytes of the address. An IPv4 address of 192.168.1.12 can be represented in IPv6 format as ::192.168.1.12, where :: represents a string of zeroes to pad the address to 128 bits.
Address Types
IPv6 introduces a number of special address types. Steel-Belted Radius accepts all equivalent forms of IPv6 addresses and recognizes them as being equivalent. Steel-Belted Radius usually displays an IPv6 address in abbreviated form, though IPv6 addresses should always be written out in full and unabbreviated form when wild cards are used. Embedded IPv4 addresses must also be represented in unabbreviated hexadecimal form when wild cards are used.

**NOTE:** The use of wild cards in IPv6 addresses is an experimental Steel-Belted Radius feature, and is not known to be documented or prohibited by any IPv6 standards at this time.

Among the types of address introduced for IPv6 addresses are the following:

- **The IPv6 unspecified address** (::) is a special address consisting entirely of zeros. The IPv6 unspecified address is analogous to the IPv4 unspecified address 0.0.0.0. You can use the unspecified address when any inbound or outbound interface is acceptable for performing a network transport function. If an inbound unspecified address is used, the server listens on all available local interfaces. If an outbound unspecified address is used, the server attempts to select an appropriate interface. The unspecified address is not routable on the larger network; a server must convert it to a routable address.

- **The IPv6 loopback address** (::1) consists of a series of zeros except for the last bit, which has a value of 1. The IPv6 loopback address, which is analogous to the IPv4 loopback address (127.0.0.1), is used to create a local network connection in memory (without involving any networking hardware), in cases where a physical network connection is not desired, in cases where the local IP address has not yet been assigned, in cases where the local IP address is not yet known, and in cases involving trouble-shooting. The loopback address is not routable on the larger network.

- **IPv6 link local unicast addresses** are used to enable IPv6 on the local network segment without requiring any IPv6 routers or IPv6 services such as Domain Name Service (DNS) or Dynamic Host Configuration Protocol (DHCP). The first 10 bits of an IPv6 link local unicast address is always 1111 1110 00 (fe80::/10 in hexadecimal form). The next 54 bits is usually zeros, followed by an interface identifier (derived from a device’s MAC address in the case of Ethernet adapters). Link local addresses can be used to contact IPv6 nodes on the local network segment. Routers do not forward packets that contain link local addresses, so link local addresses cannot be used to contact IPv6 nodes beyond the local network segment.

Link local addresses are best suited for network discovery mechanisms at OS boot time. The RFCs discourage use of link local applications in other applications, since an automatically generated link local address may not be unique beyond the local network segment.
- **IPv6 site local unicast addresses** are used to communicate with other IPv6 nodes on a local area network. The first 10 bits of an IPv6 site local unicast address is always 1111 1110 11 (fec::/10 in hexadecimal form). The next 38 bits is usually zeros, followed by a 16-bit subnet ID field. While site local addresses can be used to contact IPv6 nodes on the local area network (as well as nodes on the local network segment), site local addresses cannot be used to contact IPv6 nodes outside the local area network.

Site local addresses can be non-unique in situations where a multi-homed host is connected with more than one unrelated network. In such cases, a scope ID must be used to disambiguate non-unique site local addresses. For information on scope IDs, see “Scope IDs” on page 292. Site local addressing is likely to be deprecated as the IPv6 standard evolves.

- **IPv6 aggregatable global unicast addresses** (for example, 2:*:*:*:*:*:*:* or 3:*:*:*:*:*:*:*:*) are used to communicate with other IPv6 nodes on the Internet. Global addresses are often derived from a prototypical router address or dynamically assigned using IPv6 services such as DHCP. Global addresses can be used to contact any reachable IPv6 node on the Internet. However, not all IPv6 nodes are reachable, since some nodes use only site local or link local addresses within private networks.

- **IPv6 IPv4 mapped addresses** (for example, ::ffff:192.168.10.12) are used by IPv6-enabled applications to communicate with IPv4 nodes transparently through the IPv6 stack by means of IPv6 APIs. The IPv4 address of the IPv4 node is embedded in the last 32 bits of the IPv6 IPv4 mapped address. IPv6 IPv4 mapped addresses are routable on the Internet.

  The IPv6 stack should convert the mapped address into an IPv4 address and pass the communication through the IPv4 stack. However, IPv6 IPv4 mapped addresses are not supported on all platforms (for example, Windows XP), and this address type may be deprecated as the IPv6 standard evolves.

- **IPv6 IPv4 compatible addresses** (for example, ::192.168.21.45) are used by IPv6-enabled applications to tunnel IPv6 packets over an IPv4 routing infrastructure dynamically. The actual IPv4 address of the IPv4 node that supports a tunnel end-point is embedded in the last 32 bits of the IPv6 IPv4 compatible address. IPv6 IPv4 compatible addresses are routable on the Internet.

- **IPv6 multicast addresses** (for example, ff:*:*:*:*:*:*:*:*) are used to transmit data simultaneously to multiple IPv6 nodes. Receiving nodes subscribe to routers to receive multicast transmissions. In turn, downstream routers subscribe to upstream routers until the transmitting application is reached as the source. This address type is routable on the Internet.

- **IPv6 6to4 addresses** (for example, 2002:*:*:*:*:*:*:*) are used to provide tunneled unicast connectivity between IPv6 sites and nodes across the IPv6 Internet by treating the wide area IPv4 network as a unicast point-to-point link layer. Hosts using 6to4 addressing do not require special configuration; rather, 6to4 routers encapsulate and decapsulate IPv6 traffic in IPv4 packets.

For more information on IPv6 6to4 addresses, see RFC 3056, “Connection of IPv6 Domains via IPv4 Clouds.”
IPv6 ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) addresses

allow IPv6-in-IPv4 tunnels to be created automatically within a site. ISATAP addressing provides unicast IPv6 connectivity between IPv6 hosts across an IPv4 intranet. ISATAP treats a site’s IPv4 infrastructure as a Non-Broadcast Multiple Access (NBMA) link layer. ISATAP addresses use the locally administered interface identifier ::0:5EFE:w.x.y.z, where:

- The 0:5EFE portion is formed from the combination of the Organizational Unit Identifier (OUI) that is assigned to the Internet Assigned Numbers Authority (IANA) (00-00-5E) and a type that indicates an embedded IPv4 address (FE).
- The w.x.y.z portion is any unicast IPv4 address, which includes both public and private addresses.

The ISATAP interface identifier can be combined with any 64-bit prefix (including 6to4 prefixes) for IPv6 unicast addresses.

IPv6 Teredo addresses

allow nodes located behind an IPv4 NAT to obtain IPv6 unicast connectivity by tunneling packets over UDP/IPv4. Running the service requires the help of Teredo servers and Teredo relays:

- Teredo servers are stateless, and only have to manage a small fraction of the traffic between Teredo clients.
- Teredo relays act as IPv6 routers between the Teredo service and the "native" IPv6 Internet; the relays can also provide interoperability with hosts using other transition mechanisms such as "6to4."

If the NAT supports UDP port translation, the NAT typically supports Teredo addressing.

Scope IDs

Some IPv6 address types are designed to be globally unique, while other types of addresses, such as link local addresses or site local addresses, are limited to a local area network or a local network segment. This creates a potential ambiguity with respect to choice of network interface, especially on multi-homed hosts. For example, a local host might be assigned the same link local address on multiple network interfaces connected to different subnets. In such cases, an extra parameter, called a scope ID or zone ID, must be supplied by the application to disambiguate the addresses.

The scope ID is a non-negative integer that is attached to an IPv6 address to identify a particular network interface on a local host. For link local addresses, the scope ID is the interface number, as displayed in the `ipv6 if` command. For site-local addresses, the scope ID is the site number, as displayed in the `ipv6 if` command. For example, the following address has a scope ID of 4:

```
FE80:0:0:0:260:8FF:FEFF:FFFF%4
```

Note that a percent sign (%) is used as a delimiter between the IPv6 address and the scope ID.
At present, scope IDs are used on Windows hosts, while Solaris and Linux continue to use traditional means of specifying a particular network interface.

**Address Prefixes**

Like IPv4 addresses, IPv6 addresses are composed of a routable network number, known as the IPv6 prefix, and a host identifier, known as the IPv6 interface ID. IPv6 does not support address classes; IPv6 uses Classless Inter-Domain Routing with variable length network numbers, or prefixes, meaning that an IPv6 prefix is specified by supplying a bit length in conjunction with the address.

IPv6 prefixes are written as an IPv6 address, followed by a slash and the bit length of the prefix portion of the address. The prefix can be 0–128 bits in length, but the prefix is always written in terms of a 128-bit address. When writing prefixes, the trailing bits of the address comprising the interface ID are sometimes dropped so that the prefix can be abbreviated. The following prefixes are equivalent, assuming that the interface ID portion of the address may be ignored:

- **Canonical form:** 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64
- **Abbreviated form:** 2001:1c44:820d::/64

In many cases, the interface ID portion of the address contains relevant information. A hierarchy of prefixes may reflect the assignment and reassignment of blocks of addresses to progressively smaller organizations. For example, in a typical hierarchy, the largest service providers are assigned the largest blocks of addresses and hence the shortest prefixes, called **Top Level Aggregator Identifiers (TLA IDs)**. The large service providers reassign blocks of addresses to smaller service providers by adding a few more bits after the TLA ID; these added bits are known as **Next Level Aggregator IDs**. The smaller service providers again reassign smaller blocks of addresses to end-user organizations by again adding a few more bits after the NLA ID. These added bits are known as **Site Level Aggregator IDs**. The hierarchy continues in this way until the prefix is exhausted, leaving only the trailing bits that correspond to the non-routable IPv6 interface ID.

Steel-Belted Radius accepts all equivalent forms of IPv6 prefixes and recognizes them as being equivalent. The following prefixes are related by hierarchy, but only the canonical and abbreviated forms are equivalent:

- **Canonical form:** 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64
- **Abbreviated form:** 2001:1c44:820d::/64
- **Site level prefix:** 2001:1c44:820d::/48
- **Next level prefix:** 2001:1c44:8200::/40
- **Top level prefix:** 2001:1c00::/24

IPv6 prefixes should always be written out in full and unabbreviated form when wild cards are used, as the abbreviations become ambiguous in the presence of wild cards. The following prefixes are equivalent:

- **Canonical form:** 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64
With wild cards: 2001:1c*:??0d:eea0::*:*::*/*64

**Address Interface IDs**

Because the overall size of the IPv6 address is fixed, a longer address prefix means a shorter interface ID. For example, a 48-bit prefix implies an 80-bit interface ID:


Abbreviated prefix: 2001:1c44:820d::/48

48-bit interface ID: eea0:0260:08ff:0000:0000

Though the interface ID portion of an IPv6 address can be 0–128 bits in length, the RADIUS standard assumes 64-bit interface IDs. Steel-Belted Radius uses a convention that all interface IDs are written as a series of four unabbreviated hexadecimal fields regardless of how they are entered:

64-bit interface ID: 0260:08ff:0000:0000

IPv6 interface IDs should always be written out in full and unabbreviated form when wild cards are used, as the abbreviations become ambiguous in the presence of wild cards. The following interface IDs are equivalent:

64-bit interface ID: 0260:08ff:0000:0000

With wild cards: 02??:08ff:::*

---

**NOTE:** The use of wild cards in IPv6 interface IDs is a Steel-Belted Radius feature. Wildcards in IPv6 interface IDs are not known to be documented or prohibited by any IPv6 standards at this time.

---

**IPv6 Network Numbers**

In very specific cases, such as checklist processing, Steel-Belted Radius recognizes both IPv4 and IPv6 addresses as representing entire *ranges* of addresses. Steel-Belted Radius extends the concept of IPv4 network numbers to IPv6 as a means of representing a range of network addresses. Note that using this concept of network numbers means you cannot specify a valid network address that also happens to be a network number.

Prior to the adoption of Classless Inter-Domain Routing (CIDR), the IPv4 address space is divided into five address classes, as shown in Table 48.

**Table 48: IPv4 Address Classes**

<table>
<thead>
<tr>
<th>Class</th>
<th>Address Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0.0.0 – 127.255.255.255</td>
<td>1-bit class, 7-bit network, 24-bit host</td>
</tr>
<tr>
<td>B</td>
<td>128.0.0.0 – 191.255.255.255</td>
<td>2-bit class, 14-bit network, 16-bit host</td>
</tr>
<tr>
<td>C</td>
<td>192.0.0.0 – 223.255.255.255</td>
<td>3-bit class, 21-bit network, 8-bit host</td>
</tr>
<tr>
<td>D</td>
<td>224.0.0.0 – 239.255.255.255</td>
<td>4-bit class, 28-bit multicast group</td>
</tr>
<tr>
<td>E</td>
<td>240.0.0.0 – 247.255.255.255</td>
<td>5-bit class, 27-bit reserved</td>
</tr>
</tbody>
</table>

294   About IPv6
Within an IPv4 address class, each network is identified by a network number that consists of the leading class bits and the network bits that follow. Network numbers are typically written as IP addresses with trailing zero bits; for example, the network number corresponding to the class C address 199.100.10.24 would typically be written as 199.100.10.0.

Each network represents a potential physical interconnection of a maximum number of hosts determined by the number of host bits. Thus, the physical network identified by the network number 199.100.10.0 might connect up to 256 hosts identified by the addresses 199.100.10.0 through 199.100.10.255 inclusive. (In practice, host addresses such as 199.100.10.0 are avoided to prevent confusion between host addresses and network numbers.) Thus, it is reasonable to interpret a network number as the entire range of addresses sharing the same network portion of the address:

Network number: 199.100.10.0
Start of address range: 199.100.10.0
End of address range: 199.100.10.255
As a wild carded address: 199.100.10.*

To see how the concept of network numbers can be extended to IPv6 addresses, consider that IPv6 addresses can contain embedded IPv4 addresses. The IPv6 address ::ffff:199.100.10.0 can therefore be interpreted as the range ::ffff:199.100.10.0 through ::ffff:199.100.10.255 inclusive.

The IPv6 address space is not divided into classes, because IPv6 was designed with CIDR in mind. Constructing an arbitrary definition of IPv6 network numbers that both resembles IPv4 and scales well across all possible IPv6 addresses is difficult. However, since large portions of the IPv6 address space have not yet been defined and since the RADIUS specification concerns itself only with 64-bit interface IDs, we can consider arbitrarily assigning special meaning to all IPv6 addresses ending in 64 bits of zero. This represents a fraction (1/2^64) of the IPv6 address space, where the addresses have arbitrarily been assigned special meaning overriding their true meaning. The cases where this arbitrary definition would cause trouble are expected to be extremely rare, and it should be possible to avoid them.

Steel-Belted Radius artificially defines the concept of IPv6 network numbers as IPv6 addresses ending in 64 bits of zero, where the network number is interpreted as the entire range of IPv6 addresses sharing the same 64-bit prefix as the network number:

Network number: 2001:1c44:820d:ee0:0000:0000:0000:0000
Start of address range: 2001:1c44:820d:ee0:0000:0000:0000:0000
End of address range: 2001:1c44:820d:ee0:ffff:ffff:ffff:ffff
As a wild carded address: 2001:1c44:820d:ee0:::*::*:*:*
**IPv6 Support in Steel-Belted Radius**

In general, IPv6 support in Steel-Belted Radius parallels IPv4 support, both in terms of IPv6 network transport and in terms of RADIUS IPv6 attributes. When IPv6 networking is not supported by an operating system (either because the operating system cannot support IPv6 or because the IPv6 stack for the operating system has not been configured), Steel-Belted Radius recognizes IPv6 addresses and attributes, but does not use IPv6 transport mechanisms.

Socket interfaces in Steel-Belted Radius are both IPv4- and IPv6-capable. By default, IPv4 support is enabled and IPv6 support is disabled in Steel-Belted Radius. You must explicitly enable IPv6 support (by modifying settings in the [IPv6] section of the `radius.ini` file) before you can use IPv6 networking. Steel-Belted Radius recognizes IPv6 attributes whether or not IPv6 networking is enabled.

With few exceptions, IPv6 addresses may be configured wherever you can configure IPv4 addresses in configuration files and in the SBR Administrator.

Similarly, IPv6 RADIUS attributes can be configured wherever IPv4 RADIUS attributes can be configured. All IPv6 attributes are defined in the `radius.dct` file to allow inclusion in all standards-conforming dictionaries. IPv6 attributes are correctly interpreted and fully validated by the LDAP Configuration Interface (LCI) and by the SBR Administrator.

Table 49 presents a summary of IPv6 support in Steel-Belted Radius.

**Table 49: IPv6 Feature Matrix**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server networking</td>
<td>Yes</td>
<td>IPv6 networking must be explicitly enabled. IPv6 attributes can be processed even if IPv6 networking is not enabled.</td>
</tr>
<tr>
<td>Server DNSv6</td>
<td>Yes</td>
<td>DNSv6 must be explicitly enabled. Both IPv6 and IPv4 network connections are supported with remote DNSv6 servers. Only IPv4 network connections are supported with remote DNS servers.</td>
</tr>
<tr>
<td>Server logs</td>
<td>Yes</td>
<td>The diagnostic logging and tracing of IPv6 network connections and IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>LCI networking</td>
<td>Yes</td>
<td>If IPv6 networking is enabled, IPv6 addresses can be configured in the [LDAPAddresses] section of the <code>radius.ini</code> file.</td>
</tr>
<tr>
<td>LCI inputs</td>
<td>Yes</td>
<td>In most cases, IPv6 values can be supplied wherever IPv4 inputs can be specified.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Yes</td>
<td>Basic IPv6 attributes defined in RFC-3162 and listed in <code>radius.dct</code> are supported as native types or as regular text strings, as appropriate.</td>
</tr>
<tr>
<td>Checklists</td>
<td>Partial</td>
<td>IPv6 attributes can appear in checklists, and IPv6 address values can contain network numbers similar to IPv4 address values. IPv6 prefix values and IPv6 interface values cannot be masked or wildcarded.</td>
</tr>
<tr>
<td>Return lists</td>
<td>Yes</td>
<td>IPv6 attributes can appear within return lists, and IPv6 values can be assigned.</td>
</tr>
<tr>
<td>Attribute value pools</td>
<td>Yes</td>
<td>IPv6 attributes can appear in attribute value pools, and IPv6 values can be assigned to implement round-robin return list processing.</td>
</tr>
</tbody>
</table>
### Table 49: IPv6 Feature Matrix (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute filtering</td>
<td>Yes</td>
<td>IPv6 attributes can appear in filter rules, and IPv6 values can be assigned.</td>
</tr>
<tr>
<td>Service type mapping</td>
<td>Yes</td>
<td>IPv6 attributes can appear in a service type mapping, and IPv6 values can be wildcarded similar to IPv4 values. Reliable string comparison of regular expressions requires all values to be expressed in canonical form.</td>
</tr>
<tr>
<td>Attribute mapping</td>
<td>Yes</td>
<td>IPv6 attributes can appear in an attribute mapping, and IPv6 values can be wildcarded similar to IPv4 values. Reliable string comparison of regular expressions requires all values to be expressed in canonical form.</td>
</tr>
<tr>
<td>Class attribute</td>
<td>Partial</td>
<td>The RADIUS Class attribute cannot contain any IPv6 attributes. You can configure IPv6 addresses in the [Hosts] section of the spi.ini file to process class attributes originating from IPv6 network connections.</td>
</tr>
<tr>
<td>DHCP</td>
<td>No</td>
<td>The use of IPv6 networking to communicate with any DHCP server is not supported. The allocation of IPv6 addresses obtained from any DHCP server is not supported.</td>
</tr>
<tr>
<td>IP address pools</td>
<td>No</td>
<td>The allocation of IPv6 addresses from an SBR-managed IP address pool is not supported. However, RFC 3162 provides an attribute, Framed-IPv6-Pool, that allows the RAS to implement an IPv6 address pool.</td>
</tr>
<tr>
<td>Networking for authentication</td>
<td>Yes</td>
<td>IPv6 addresses can be configured in the [Addresses] section of the radius.ini file.</td>
</tr>
<tr>
<td>Authentication logs</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Local User authentication</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Authenticate-Only requests</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Pass-through authentication</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is not necessarily supported with external services such as RSA SecurID and TACACS+.</td>
</tr>
<tr>
<td>External authentication</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is not necessarily supported with external services such as LDAP and SQL.</td>
</tr>
<tr>
<td>EAP-TTLS authentication</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Directed authentication</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Networking for accounting</td>
<td>Yes</td>
<td>IPv6 addresses can be configured in the [Addresses] section of the radius.ini file.</td>
</tr>
<tr>
<td>Accounting logs</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>External accounting (for example, SQL)</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is not necessarily supported with external services such as SQL.</td>
</tr>
<tr>
<td>Directed accounting</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
</tbody>
</table>
### Table 49: IPv6 Feature Matrix (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spooled accounting</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported. IPv6 addresses can be configured in the [Interfaces] section of the <code>proxy.ini</code> file. Both classic and extended proxy support IPv6.</td>
</tr>
<tr>
<td>Networking for proxy</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported. Both classic and extended proxy support IPv6.</td>
</tr>
<tr>
<td>Proxy authentication</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported. Both classic and extended proxy support IPv6.</td>
</tr>
<tr>
<td>Proxy accounting</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported. Both classic and extended proxy support IPv6.</td>
</tr>
<tr>
<td>3GPP2</td>
<td>Partial</td>
<td>RFC standards are not sufficiently evolved to enable full support of IPv6. Although IPv6 attributes can be processed, they are not meaningful in the context of 3GPP2.</td>
</tr>
<tr>
<td>Master SNMP agent</td>
<td>No</td>
<td>(Solaris/Linux only) The use of IPv6 networking to communicate with an IPv6 capable SNMP management station and/or SNMP subagent is not supported.</td>
</tr>
<tr>
<td>SNMP subagent</td>
<td>No</td>
<td>(Solaris/Linux only) IPv6 networking, IPv6 trap variables, and IPv6 MIB objects are not supported. IPv6 addresses are reported as IPv4 MIB objects possessing the value 255.255.255.255.</td>
</tr>
<tr>
<td>Windows events</td>
<td>No</td>
<td>(Windows only) Neither IPv6 networking nor IPv6 event variables are supported at this time.</td>
</tr>
<tr>
<td>Networking for plug-Ins</td>
<td>Yes</td>
<td>Steel-Belted Radius does not control the networking of back end servers with its plug-ins. IPv6 networking is generally a hidden detail of third-party back end server configuration.</td>
</tr>
<tr>
<td>Plug-In APIs</td>
<td>Yes</td>
<td>IPv6 features and parameters are exposed in the new plug-in APIs. The older plug-in APIs are deprecated but still functional. You should upgrade to the new plug-in APIs to gain access to IPv6 features. IPv6 APIs can be invoked even if IPv6 networking is not enabled.</td>
</tr>
<tr>
<td>Plug-In attributes</td>
<td>Yes</td>
<td>Basic IPv6 attributes defined in RFC-3162 and listed in <code>radius.dct</code> are supported as native types or as regular text strings, as appropriate.</td>
</tr>
<tr>
<td>Oracle plug-Ins (Solaris)</td>
<td>Partial</td>
<td>(Solaris only) IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>ODBC plug-Ins</td>
<td>Partial</td>
<td>(Windows/Linux only) IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>JDBC plug-Ins</td>
<td>Partial</td>
<td>(Windows/Linux only) IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>LDAP plug-In</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>RSA SecurID plug-Ins</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>PEAP Plug-In</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported, but the required third-party software either does not currently support IPv6 or we have not tested it.</td>
</tr>
</tbody>
</table>
Table 49: IPv6 Feature Matrix (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS plug-In</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>TTLS plug-In</td>
<td>Partial</td>
<td>IPv6 attributes are fully supported, but the required third-party software may not support IPv6.</td>
</tr>
<tr>
<td>PAS plug-Ins</td>
<td>Yes</td>
<td>(SPE only) IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Concurrency plug-Ins</td>
<td>Yes</td>
<td>(SPE only) IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>Uniport Plug-In</td>
<td>Yes</td>
<td>IPv6 attributes are fully supported.</td>
</tr>
<tr>
<td>3COM CCA Tunnels (deprecated)</td>
<td>Partial</td>
<td>The use of IPv6 networking is not supported. IPv6 attributes can be processed even if IPv6 networking is not enabled.</td>
</tr>
</tbody>
</table>

RADIUS IPv6 Attributes

All RADIUS attributes defined in RFC 3162, RADIUS and IPv6, are supported in Steel-Belted Radius as native types or as regular text strings. All forms of attribute processing, such as checklist processing, return list processing, attribute echoing/deleting/merging, are supported. However, IPv6 prefix values and IPv6 interface values cannot be masked or wildcarded in checklist processing.

Third-party plug-ins that have not been upgraded to support IPv6 should be able to process IPv6 attributes as opaque hexadecimal strings.

Table 50 lists the attribute number, and the number of times an attribute can appear in an Access-Request, Access-Accept, Access-Reject, Access-Challenge, and Accounting-Request packets for each type of IPv6 RADIUS attribute.

Table 50: IPv6-Specific RADIUS Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attr Num</th>
<th>Acc-Req</th>
<th>Acc-Accept</th>
<th>Acc-Rej</th>
<th>Acc-Chall</th>
<th>Acc-Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAS-IPv6-Address</td>
<td>95</td>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0-1</td>
</tr>
<tr>
<td>Framed-Interface-Id</td>
<td>96</td>
<td>0-1</td>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0-1</td>
</tr>
<tr>
<td>Framed-IPv6-Prefix</td>
<td>97</td>
<td>0+</td>
<td>0+</td>
<td>0</td>
<td>0</td>
<td>0+</td>
</tr>
<tr>
<td>Login-IPv6-Host</td>
<td>98</td>
<td>0+</td>
<td>0+</td>
<td>0</td>
<td>0</td>
<td>0+</td>
</tr>
<tr>
<td>Framed-IPv6-Route</td>
<td>99</td>
<td>0</td>
<td>0+</td>
<td>0</td>
<td>0</td>
<td>0+</td>
</tr>
<tr>
<td>Framed-IPv6-Pool</td>
<td>100</td>
<td>0</td>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0-1</td>
</tr>
</tbody>
</table>

NAS-IPv6-Address

The NAS-IPv6-Address attribute is similar in function to the NAS-IP-Address attribute. Either attribute is sufficient to identify the IP address of the requesting RAS to the RADIUS server. If both attributes appear in the same RADIUS Access-Request packet, Steel-Belted Radius processes the NAS-IPv6-Address attribute for the purpose of identifying the RAS.
The NAS-IPv6-Address attribute may be specified by the RAS in access and accounting request packets. Zero or one NAS-IPv6-Address attributes may be specified. If present, the fixed length NAS-IPv6-Address attribute contains the complete 128-bit IPv6 address of the requesting RAS.

Steel-Belted Radius allows zero or one 128-bit IPv6 address to be specified for each RAS. The server authentication logic validates these addresses on extraction from the database and compares them with NAS-IPv6-Address attributes when they are received in access and accounting request packets. The server accounting logic writes these addresses to the accounting logs in a human readable format.

Example

Human readable: fe80::260:8ff:feff:ffff
RADIUS attribute: 5f 12 fe 80 00 00 00 00 02 60 08 ff fe ff ff

Framed-Interface-Id

The Framed-Interface-Id attribute specifies the IPv6 interface ID to be assigned to a user. When combined with a Framed-IPv6-Prefix attribute, a single Framed-Interface-Id attribute forms one or more complete IPv6 addresses to be assigned to the user.

In general, the user is assigned the number of addresses equal to the number of Framed-IPv6-Prefix attributes, where the addresses have the same Framed-Interface-Id value and different Framed-IPv6-Prefix values.

It is possible to assign complete IPv6 addresses using only Framed-IPv6-Prefix attributes (i.e. without specifying any Framed-Interface-Id attribute). For example, in the case of PPP, it can be quite difficult to automatically generate a unique IPv6 interface ID for a given network segment, so it is recommended that the RADIUS server honor the hint if this attribute is suggested by the RAS. This is typically accomplished with echo attributes.

The Framed-Interface-Id attribute may be specified by the RAS in Access- and Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or one Frame-Interface-Id attributes may be specified. If present, the fixed length Framed-Interface-Id attribute contains the 64-bit interface ID to be assigned to the user.

Steel-Belted Radius allows zero or one 64-bit interface ID to be specified for each local user. The server authentication logic validates this interface ID on extraction from the database and includes the Framed-Interface-Id attribute in Access-Accept packets if none is received in the Access-Request packets.

Example

Human readable: 260:8ff:feff:ffff
RADIUS attribute: 60 0a 02 60 08 ff fe ff ff

Framed-IPv6-Prefix

The Framed-IPv6-Prefix attribute specifies the IPv6 networks to be assigned to a user. When combined with a Framed-Interface-ID attribute, a single Framed-IPv6-Prefix attribute forms one or more complete IPv6 addresses to be assigned to the user.
In general, the user is assigned the number of addresses equal to the number of Framed-IPv6-Prefix attributes, where the addresses have the same Framed-Interface-Id value, but different Framed-IPv6-Prefix values.

It is possible to assign complete IPv6 addresses using only Framed-IPv6 Prefix attributes (that is, without specifying any Framed-Interface-Id attribute). For example, the Framed-IPv6-Prefix attributes may be suggested by the RAS and overridden by the RADIUS server. In any case, the RAS is expected to be able to plumb the routes implied by the Framed-IPv6-Prefix attributes and these need not be repeated in separate Framed-IPv6-Route attributes.

The Framed-IPv6-Prefix attribute may be specified by the RAS in Access- and Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or more Framed-IPv6-Prefix attributes may be specified. If present, the variable-length Framed-IPv6-Prefix attribute contains an IPv6 prefix from 0 to 128 bits in length. Extra bits beyond the actual prefix length must be set to 0.

Steel-Belted Radius allows zero or more variable-length IPv6 prefixes to be specified for each local user or attribute profile. The server authentication logic validates these prefixes on extraction from the database and includes the proper number of Framed-IPv6-Prefix attributes in Access-Accept packets if none are received in the access request packets. The server accounting logic writes these prefixes to the accounting logs in a human readable format.

**Example**

Human readable:fe80::260:8ff:feff:ffff/10
RADIUS attribute:61 14 00 0a fe 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(8-bit type) (8-bit length)
(8-bit zero) (8-bit prefix length) (128-bit IPv6 prefix)

**Login-IPv6-Host**

The Login-IPv6-Host attributes specify the IPv6 addresses of the systems with which the user is connected when the Login-Service attribute is also included. The Login-IPv6-Host attribute may be suggested by the RAS and overridden by the RADIUS server.

The Login-IPv6-Host attribute may be specified by the RAS in access and accounting request packets, and by the RADIUS server in Access-Accept packets. Zero or more Login-IPv6-Host attributes may be specified. If present, the fixed length Login-IPv6-Host attribute contains the complete 128-bit IPv6 address of the login host, or a special value:

- 128-bits set to 0 indicates that the RAS should select the login host for the user.
- 128-bits set to 1 indicates that the RAS should allow the user to select the login host.
- Other values indicate the actual 128-bit IPv6 address of the login host.
Steel-Belted Radius allows zero or more 128-bit IPv6 addresses (including special values) to be specified for each local user or attribute profile. The server authentication logic validates these addresses (including special values) on extraction from the database and includes the proper number of Login-IPv6-Host attributes in Access-Accept packets if none are received in the access request packets. The server accounting logic writes these addresses to the accounting logs in a human readable format.

**Example**

Human readable: `fe80::260:8ff:feff:ffff`

RADIUS attribute: `62 12 fe 80 00 00 00 00 00 02 60 08 ff fe ff ff ff`

**Framed-IPv6-Pool**

The Framed-IPv6-Pool attribute specifies the name of a RAS managed pool (as opposed to a RADIUS server managed pool) from which the RAS should assign an IPv6 prefix to the user. The Framed-IPv6-Pool attribute may not be suggested by the RAS and is always determined by the RADIUS server.

The Framed-IPv6-Pool attribute may be specified by the RAS in Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or one Framed-IPv6-Pool attributes may be specified. If present, the variable-length Framed-IPv6-Pool attribute contains the name of a RAS managed pool in human readable text. The text is not NULL terminated.

Steel-Belted Radius allows zero or one variable length pool name to be specified for each local user. The server authentication logic validates the pool name on extraction from the database and includes the proper number of Framed-IPv6-Pool attributes in Access-Accept packets. The server accounting logic writes these pool names to the accounting logs in a human readable format.

**Example**

Human readable: `ipv6-pool`

RADIUS attribute: `64 0b 69 70 76 36 2d 70 6f 6f 6c`

**Framed-IPv6-Route**

The Framed-IPv6-Route attribute specifies the IPv6 routing information to be configured for the user on the RAS. The RAS is expected to be able to plumb the routes specified by the Framed-IPv6-Route attributes in addition to those that may already be implied by separate Framed-IPv6-Prefix attributes. The Framed-IPv6-Route attribute may not be suggested by the RAS and is always determined by the RADIUS server.

The Framed-IPv6-Route attribute may be specified by the RAS in accounting request packets, and by the RADIUS server in Access-Accept packets. Zero or more Framed-IPv6-Route attributes may be specified. If present, the variable-length Framed-IPv6-Route attribute contains IPv6 routing information in human readable text. The text is not NULL terminated. The format of the text (destination prefix, gateway address, metrics) is described in RFC-3162.
Steel-Belted Radius allows zero or more variable-length IPv6 routes to be specified for each local user or attribute profile. The server authentication logic validates these routes on extraction from the database and includes the proper number of Framed-IPv6-Route attributes in Access-Accept packets. The server accounting logic writes these routes to the accounting logs in a human readable format.

**Example**

Human readable:2000:0:0:106::/64 2000::106:a00:20ff:fe99:a998 1  
RADIUS attribute:63 32 32 30 30 30 3a 30 3a 30 ... 39 3a 61 39 39 38 20 31

---

**Configuring IPv6**

This section describes how to configure Steel-Belted Radius to use IPv6 networking.

**Enabling IPv6 Networking**

To enable IPv6 networking in Steel-Belted Radius, you must modify the `radius.ini` file and then restart your Steel-Belted Radius server. For information on the settings in the `radius.ini` file, refer to the Steel-Belted Radius Reference Guide.

Note that Steel-Belted Radius can process IPv6 attributes even if IPv6 networking is not enabled, provided that the IPv6 attributes are described in the RADIUS dictionary files.

**Configuring IPv6 Scope IDs**

Some types of IPv6 addresses require an IPv6 scope ID to avoid address ambiguity. In some cases, the Steel-Belted Radius server can select a scope ID automatically. You can specify the scope ID explicitly for IPv6 link local and site local addresses.

The `[IPv6]` section of the `radius.ini` file can specify how scope IDs are selected for each IPv6 address type that is recognized by the server. If the parameter value is 0, the Steel-Belted Radius server selects a scope ID automatically. If the parameter value is not 0, then the Steel-Belted Radius server uses that value as the scope ID when establishing network connections involving that IPv6 address type.

**NOTE:** You can use the output of the `ifconfig -a` shell command on Solaris/Linux platforms, and the output of the `ipconfig /all` shell command on Windows platforms (`ipv6 if` on Windows XP platforms) to determine the proper host specific scope ID for an address type. The scope ID is identical to the interface index on which the address type is supported and on which the desired destinations are reachable. On Solaris/Linux platforms, the server accepts traditional interface names, such as `hme0`, instead of numeric scope IDs.
Configuring IPv6 Addresses for RADIUS Client Connections

You can configure the [Addresses] section of the `radius.ini` file if you want to specify the local address or addresses on which Steel-Belted Radius listens for RADIUS client connections. By default, Steel-Belted Radius automatically discovers and configures all available IPv4 interfaces on the local host. If IPv6 is enabled, Steel-Belted Radius discovers and configures both IPv4 and IPv6 interfaces.

You can configure Steel-Belted Radius to configure IPv4 automatically by entering `AutoConfigureIPv4` in the [Addresses] section. Similarly, you can configure Steel-Belted Radius to configure IPv6 automatically by entering `AutoConfigureIPv6` in the [Addresses] section. If you configure specific IPv4 or IPv6 addresses, Steel-Belted Radius listens for RADIUS traffic on only those interfaces.

Steel-Belted Radius automatically uses site local addresses if autoconfiguration is enabled, and allows site local addresses to be configured manually. Unless a single shared scope ID is also configured for these addresses (that is, unless all site local addresses exist on the same subnet), Steel-Belted Radius tries to determine the appropriate scope ID for any given site local address automatically.

The IPv6 unspecified address `::` allows connections on any IPv6 address or IPv4 address, with IPv4 connections represented as IPv6 IPv4 mapped addresses. Because IPv6 IPv4 mapped addresses are not currently supported by the Windows IPv6 protocol stack, you must enter the IPv4 unspecified address `0.0.0.0` with the IPv6 unspecified address `::` to approximate the desired behavior on Windows platforms.

Steel-Belted Radius does not use link local addresses unless they are configured explicitly. To use link local addresses, you must configure a nonzero `IPv6LinkLocalUnicastScopeId` in the [IPv6] section of the `radius.ini` file. If configured, all link local addresses must exist on the same network interface and use the same scope ID.

Configuring DNSv6 Support

Enabling Domain Name Service Version 6 (DNSv6) support lets Steel-Belted Radius communicate with a DNSv6 server to resolve host names. By default, Steel-Belted Radius uses DNS unless IPv6 is enabled and DNSv6 support is configured by means of the `DynamicNameResolution` parameter in the [IPv6] section of the `radius.ini` file.

- If `DynamicNameResolution` is set to 0, Steel-Belted Radius uses IPv4 DNS, which means it does not query DNSv6 services.
- If `DynamicNameResolution` is set to 1, Steel-Belted Radius uses IPv6 DNS (DNSv6), which means it does not query IPv4 DNS services and ignores IPv4-specific information returned by DNSv6 services.
- If `DynamicNameResolution` is set to 2, Steel-Belted Radius queries DNSv6 services for IPv6-specific information, and then queries IPv4 DNS services for IPv4 specific information if DNSv6 fails to resolve a host name.
Appendix G

Stopping and Starting Steel-Belted Radius

This appendix describes how to stop and restart the Steel-Belted Radius server.

Stopping the Steel-Belted Radius Server

After the Steel-Belted Radius service (Windows) or RADIUS daemon (Solaris/Linux) is installed, it stops and starts automatically each time you shut down or restart the server. If you modify the settings in the Steel-Belted Radius configuration files, you may need to restart the Steel-Belted Radius server manually before the server recognizes its new settings.

Windows

You can stop the Steel-Belted Radius service at any time by performing the following steps:


2. When the Services window (Figure 133) opens, click the Steel-Belted Radius entry.
3. Click the **Stop the service** button.

**Solaris**

You can stop the RADIUS daemon on a Solaris server at any time by issuing the following commands:

```
cd server-directory
./sbrd stop
```

**Linux**

After the RADIUS daemon is installed on the server, it stops and starts automatically each time you shut down or restart the server. You can stop the RADIUS daemon on a Linux server at any time by issuing the following commands:

```
cd server-directory
./sbrd stop
```

When you execute the `sbrd stop` command, Steel-Belted Radius allows its subsystems to complete outstanding work, release resources, and then stops the `mkded` (btrieve) daemon and the `radius` service gracefully.

If Steel-Belted Radius fails to stop after you issue an `sbrd stop` command, you can use the optional `force` argument to terminate all subsystems immediately.

```
cd server-directory
./sbrd stop force
```
Starting the Steel-Belted Radius Server

You must restart the Steel-Belted Radius service (Windows) or RADIUS daemon (Solaris/Linux) after you modify the Steel-Belted Radius configuration files.

**Windows**

To start the Steel-Belted Radius service on a Windows server after it has been stopped:

1. Choose **Start > Control Panel > Administrative Tools > Services.**
2. When the Services window (Figure 133 on page 306) opens, click the Steel-Belted Radius entry.
3. Click the **Start the service** button.

To restart the Steel-Belted Radius service without stopping it:

1. Choose **Start > Control Panel > Administrative Tools > Services.**
2. When the Services window (Figure 133 on page 306) opens, click the Steel-Belted Radius entry.
3. Click the **Restart the service** button.

**Solaris**

Use the following commands to start the RADIUS daemon on a Solaris server:

```
cd server-directory
./sbrd start
```

**Linux**

Use the following command to start the RADIUS daemon after you have issued an `sbrd stop` command on a Linux server:

```
cd server-directory
./sbrd start
```

When you execute the `sbrd start` command, Steel-Belted Radius starts the `mkded` (btrieve) daemon to allow database access, and then starts the `radius` service.

If you change configuration settings for your Steel-Belted Radius server, you may need to restart the server to make the changes effective. As an alternative to issuing an `sbrd stop` command immediately followed by an `sbrd start` command, you can use the `sbrd restart` command when you want to restart Steel-Belted Radius. When you issue the `sbrd restart` command, the system shuts down the `mkded` (btrieve) daemon and the `radius` service (if they are running), and then immediately starts the `mkded` (btrieve) daemon and the `radius` service.

```
cd server-directory
./sbrd restart
```
Displaying RADIUS Status Information (Linux)

You can use the `sbrd status` command to display status information for the RADIUS daemon.

```
    cd server-directory
    ./sbrd status
```

Figure 134 illustrates the output of the `sbrd status` command.

**Figure 134: Output of sbrd status Command**

```
> sbrd status
    ecarter 25927 .mkded start
    btrieve processes are active

    —— Shared Memory Segments ——
    key shmidownerpermsbytesnattchstatus
    0x42545256891968ecarter60080000002

    —— Semaphore Arrays ——
    key semidownerpermsnsems
    0x42545256167116ecarter660250

    btrieve shared IPC objects exist
    btrieve state is running
    btrieve status 1101

    ecarter 26066 radius -d/home/ecarter/sbr/5.0.5.1553/funk/radius
    sbr.xml
    radius processes are running
    radius state is running
    radius status 1101

    Aggregate state is running
```
Index

Symbols
% AuthType .......................................................... 212
% FullName ........................................................ 212
% Name ............................................................. 198
% NASAddress .................................................... 198, 212
% NASModel .......................................................... 199, 212
% NASName .......................................................... 198, 199, 212
% OriginalUserName ............................................. 198
% Password ......................................................... 199
% password .......................................................... 199
% Time .............................................................. 212
% TransactionTime ................................................. 212
% Type .............................................................. 212
% UserName ........................................................ 198

Numerics
802.1X access point ........................................... 3

A
access client ....................................................... 3
Access-Accept message ........................................ 4
Access-Reject message ......................................... 4
Access-Request message ....................................... 17
account.ini ......................................................... 239
Acct-Authentic .................................................... 254
Acct-Delay-Time ................................................. 254
Acct-Status-Type ............................................... 254
Acct-Termination-Cause ...................................... 255
address leak ....................................................... 50
address pools ..................................................... 88
attribute mapping ................................................ 18
attributes .......................................................... 5
Authenticate-Only requests ................................... 19
authentication .....................................................
inner ................................................................. 124
authentication methods list ................................... 17, 149, 197
authlog.ini .......................................................... 239
authReport.ini ..................................................... 239
authReportAccept.ini .......................................... 239
authReportBadSharedSecret.ini ............................. 239
authReportReject.ini ........................................... 239
authReportUnknownClient.ini ............................... 239
automatic EAP helper ......................................... 106
automatic EAP helpers ......................................... 106

B
Bind Authentication ............................................ 221
BindName Authentication ..................................... 221
Bootstrap section ................................................
in sidalt.aut file .................................................... 272

C
CacheTimeoutAttr ............................................... 272
challenge .......................................................... 23
ChallengTokenInPassword .................................... 273
CHAP ............................................................... 22
checklist attributes ............................................. 11
concurrent connections ........................................ 65
concurrent tunnel connections ............................... 32, 79
concurrent users .................................................. 31

D
Diffie-Hellman ................................................... 117, 123, 131, 137, 144
digest ............................................................... 23
directed authentication ........................................ 18
domain controller ............................................... 124
Dropped Packet .................................................. 232, 233

E
EAP .................................................................
EAP Identity Response .......................................... 105
EAP Setup window ............................................... 149
EAP-FAST .......................................................... 131
EAP-Message attribute ........................................ 105, 107
EAP-PEAP .......................................................... 138
EAP-TTLS ........................................................... 124
echo property ...................................................... 13
eDirectory .......................................................... 265
Enable ............................................................... 272
Enc-md5 ............................................................. 202, 226
exponentiation ................................................... 117, 123, 131, 137, 144
Extensible Authentication Protocol, see EAP ....... 25
external accounting .............................................. 25
external authentication ........................................ 18

F
Failed Authentication ......................................... 232
Failed on Checklist ............................................ 232
force ................................................................. 306
Framed-Compression ........................................... 12
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameIPAddressHint</td>
<td>28</td>
</tr>
<tr>
<td>Hint</td>
<td>28</td>
</tr>
<tr>
<td>Host agent</td>
<td>9</td>
</tr>
<tr>
<td>InitializationString</td>
<td>20, 21, 196, 272</td>
</tr>
<tr>
<td>Insufficient Resources</td>
<td>232, 234</td>
</tr>
<tr>
<td>Invalid Client</td>
<td>234</td>
</tr>
<tr>
<td>Invalid Request</td>
<td>232, 233</td>
</tr>
<tr>
<td>IP address pool</td>
<td>187</td>
</tr>
<tr>
<td>IP address pools</td>
<td>85</td>
</tr>
<tr>
<td>IP Pools tab</td>
<td>85</td>
</tr>
<tr>
<td>IPX address pool</td>
<td>187</td>
</tr>
<tr>
<td>IPX address pools</td>
<td>88</td>
</tr>
<tr>
<td>IPX Pools tab</td>
<td>88</td>
</tr>
<tr>
<td>LDAP</td>
<td>124</td>
</tr>
<tr>
<td>Log files</td>
<td>25</td>
</tr>
<tr>
<td>LogAccept</td>
<td>252</td>
</tr>
<tr>
<td>LogLevel</td>
<td>252</td>
</tr>
<tr>
<td>Make/model field</td>
<td>10</td>
</tr>
<tr>
<td>MaxConcurrent</td>
<td>200, 215</td>
</tr>
<tr>
<td>MessageID</td>
<td>202, 226</td>
</tr>
<tr>
<td>MD4</td>
<td>273</td>
</tr>
<tr>
<td>MS-CHAP</td>
<td>22</td>
</tr>
<tr>
<td>MS-CHAP-V2</td>
<td>22</td>
</tr>
<tr>
<td>MS-CHAP-v2</td>
<td>106</td>
</tr>
<tr>
<td>multi-valued attributes</td>
<td>12</td>
</tr>
<tr>
<td>mutual authentication</td>
<td>112</td>
</tr>
<tr>
<td>Native User</td>
<td>17, 49, 56</td>
</tr>
<tr>
<td>Native User authentication</td>
<td>17</td>
</tr>
<tr>
<td>NDS, see Novell eDirectory</td>
<td>17</td>
</tr>
<tr>
<td>network access device (NAD)</td>
<td>12</td>
</tr>
<tr>
<td>Next Level Aggregator IDs.</td>
<td>293</td>
</tr>
<tr>
<td>Novell eDirectory</td>
<td>265</td>
</tr>
<tr>
<td>Novell NDS</td>
<td>265</td>
</tr>
<tr>
<td>Orderable attributes</td>
<td>197, 202, 211, 216</td>
</tr>
<tr>
<td>Oracle</td>
<td>12</td>
</tr>
<tr>
<td>PannelTunnels</td>
<td>80</td>
</tr>
<tr>
<td>PAP</td>
<td>22</td>
</tr>
<tr>
<td>pass-through authentication</td>
<td>17</td>
</tr>
<tr>
<td>password output parameter</td>
<td>199</td>
</tr>
<tr>
<td>PEAP</td>
<td>138</td>
</tr>
<tr>
<td>perfmon counters</td>
<td>276</td>
</tr>
<tr>
<td>phantom records</td>
<td>32</td>
</tr>
<tr>
<td>Pool</td>
<td></td>
</tr>
<tr>
<td>IP address</td>
<td>85</td>
</tr>
<tr>
<td>Prefetch-capable</td>
<td>108</td>
</tr>
<tr>
<td>profiles</td>
<td>67</td>
</tr>
<tr>
<td>Proxy Failure</td>
<td>232, 234</td>
</tr>
<tr>
<td>proxy RADIUS</td>
<td>17</td>
</tr>
<tr>
<td>proxy RADIUS authentication</td>
<td>17</td>
</tr>
<tr>
<td>RADIUS daemon, starting and stopping</td>
<td>305, 306</td>
</tr>
<tr>
<td>RADIUS server</td>
<td>4</td>
</tr>
<tr>
<td>radius.dct</td>
<td>10</td>
</tr>
<tr>
<td>radiusclass</td>
<td>169</td>
</tr>
<tr>
<td>radiusdir</td>
<td>xiv</td>
</tr>
<tr>
<td>radsql.acc</td>
<td>209</td>
</tr>
<tr>
<td>radsql.aut</td>
<td>196</td>
</tr>
<tr>
<td>radsqjdbc.acc</td>
<td>209</td>
</tr>
<tr>
<td>redundancy</td>
<td>7</td>
</tr>
<tr>
<td>Rejected by Proxy</td>
<td>232</td>
</tr>
<tr>
<td>rejection messages</td>
<td>151</td>
</tr>
<tr>
<td>retry policy</td>
<td>73</td>
</tr>
<tr>
<td>return-list attributes</td>
<td>11</td>
</tr>
<tr>
<td>RSA SecurID</td>
<td>4, 20, 147</td>
</tr>
<tr>
<td>Sbrd restart</td>
<td>307</td>
</tr>
<tr>
<td>sbrd start</td>
<td>307</td>
</tr>
<tr>
<td>sbrd status</td>
<td>308</td>
</tr>
<tr>
<td>sbrd stop</td>
<td>306</td>
</tr>
<tr>
<td>Sbrd stop force</td>
<td>306</td>
</tr>
<tr>
<td>Sbrsetupool</td>
<td>160, 161</td>
</tr>
<tr>
<td>SecurID</td>
<td>4</td>
</tr>
<tr>
<td>see RSA SecurID</td>
<td></td>
</tr>
<tr>
<td>SecurID authentication</td>
<td>147</td>
</tr>
<tr>
<td>SecurID user</td>
<td>58</td>
</tr>
<tr>
<td>SecurID, see RSA SecurID</td>
<td></td>
</tr>
<tr>
<td>session resumption</td>
<td>110</td>
</tr>
<tr>
<td>Session-Timeout attribute</td>
<td>110</td>
</tr>
<tr>
<td>Settings section</td>
<td></td>
</tr>
<tr>
<td>in sidalt.aut file</td>
<td>272</td>
</tr>
<tr>
<td>shared secret</td>
<td>8, 9</td>
</tr>
<tr>
<td>sidalt.aut</td>
<td>272</td>
</tr>
<tr>
<td>signed integer, in attributes</td>
<td>12</td>
</tr>
<tr>
<td>Site Level Aggregator IDs.</td>
<td>293</td>
</tr>
<tr>
<td>SQL</td>
<td>124</td>
</tr>
<tr>
<td>sqlacct.acc</td>
<td>209</td>
</tr>
<tr>
<td>Term</td>
<td>Page(s)</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>sqlauth.aut</td>
<td>196</td>
</tr>
<tr>
<td>stop force</td>
<td>306</td>
</tr>
<tr>
<td>system assigned values</td>
<td>12</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>tab</td>
<td></td>
</tr>
<tr>
<td>IP Pools</td>
<td>85</td>
</tr>
<tr>
<td>IPX Pools</td>
<td>88</td>
</tr>
<tr>
<td>TACACS +</td>
<td>4, 20, 23, 148</td>
</tr>
<tr>
<td>TokenAttr</td>
<td>272</td>
</tr>
<tr>
<td>Top Level Aggregator IDs</td>
<td>293</td>
</tr>
<tr>
<td>TraceLevel</td>
<td>252</td>
</tr>
<tr>
<td>tunnel connections, concurrent</td>
<td>32, 79</td>
</tr>
<tr>
<td>tunnels</td>
<td>80, 186</td>
</tr>
<tr>
<td>Tunnels panel</td>
<td>80</td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>UNIXcrypt</td>
<td>201, 202, 225, 226</td>
</tr>
<tr>
<td>User type field</td>
<td>11</td>
</tr>
<tr>
<td>User-Name attribute</td>
<td>106</td>
</tr>
<tr>
<td>users, concurrent</td>
<td>31</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>vendor-specific attributes</td>
<td>10</td>
</tr>
</tbody>
</table>