RSA Authentication Manager 7.1
RADIUS Reference Guide

The Security Division of EMC

Contact Information

See the RSA corporate web site for regional Customer Support telephone and fax numbers**:** **www.rsa.com**

Trademarks

RSA and the RSA logo are registered trademarks of RSA Security Inc. in the United States and/or other countries. For the most up-to-date listing of RSA trademarks, see **www.rsasecurity.com/legal/trademarks_list.pdf**. EMC is a registered trademark of EMC Corporation. All other goods and/or services mentioned are trademarks of their respective companies.

License agreement

This software and the associated documentation are proprietary and confidential to RSA, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Note on encryption technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Limit distribution of this document to trusted personnel.

RSA notice

The RC5™ Block Encryption Algorithm With Data-Dependent Rotations is protected by U.S. Patent #5,724,428 and #5,835,600.

First printing: April 2008

Contents

Preface 5
About This Guide 5
RSA Authentication Manager Documentation 5
Tutorials 6
Getting Support and Service 6
Before You Call Customer Support 6

Chapter 1: radius.ini File 7
[Addresses] Section 7
Example 1 8
Example 2 8
[AuditLog] Section 8
[Configuration] Section 9
[CurrentSessions] Section 17
[Debug] Section 17
[EmbedInClass] Section 18
[FailedAuthOriginStats] Section (Windows only) 18
[HiddenEAPIdentity] Section 19
[IPv6] Section 20
[MsChapNameStripping] Section 21
[Ports] Section 22
[SecurID] Section 24
[UserNameTransform] Section 24

Chapter 2: sbrd.conf File (Solaris/Linux only) 27

Chapter 3: services File 31

Chapter 4: securid.ini File 33
[Configuration] Section 33
[Server_Settings] Section 34
[Prompts] Section 35
Substitution String Formats 35

Chapter 5: Attribute Processing Files 39
Overview 39
Dictionary File Location 40
Dictionary File Records 40
Editing Dictionary Files 41
Include Records 41
ATTRIBUTE Records 42
Macro Records 45
OPTION Records 46

classmap.ini File 47
[AttributeName] Section 47
filter.ini File 48
Filter Rules 48
Order of Filter Rules 49
Values in Filter Rules 50
Referencing Attribute Filters 51
spi.ini File 52
[Keys] Section 53
[Hosts] Section 53
vendor.ini File 54
[Vendor-Product Identification] Section 54

Chapter 6: Accounting Configuration Files 57
account.ini File 57
[Alias/name] Sections 57
[Attributes] Section 58
[Settings] Section 59
[TypeNames] Section 62

Chapter 7: EAP Configuration Files 65
eap.ini File 65
peapauth.aut File 67
[Bootstrap] Section 67
[Server_Settings] Section 68
[Session_Resumption] Section 70
ttlsauth.aut File 71
[Bootstrap] Section 71
[Server_Settings] Section 72
[Session_Resumption] Section 73
[Integrity_Settings] 74
Sample ttlsauth.aut File 75

Index 77

Preface

About This Guide

This guide describes the usage and settings for the initialization files, dictionary files, and configuration files used by RSA RADIUS.

It is intended for administrators and other trusted personnel. Do not make this guide available to the general user population.

RSA Authentication Manager Documentation

For more information about RSA Authentication Manager, see the following documentation:

Release Notes. Provides information about what is new and changed in this release, as well as workarounds for known issues.

Getting Started. Lists what the kit includes (all media, diskettes, licenses, and documentation), specifies the location of documentation on the DVD or download kit, and lists RSA Customer Support web sites.

Planning Guide. Provides a general understanding of RSA Authentication Manager, its high-level architecture, its features, and deployment information and suggestions.

Installation and Configuration Guide. Describes detailed procedures on how to install and configure RSA Authentication Manager.

Administrator's Guide. Provides information about how to administer users and security policy in RSA Authentication Manager.

Migration Guide. Provides information for users moving from RSA Authentication Manager 6.1 to RSA Authentication Manager 7.1, including changes to terminology and architecture, planning information, and installation procedures.

Developer's Guide. Provides information about developing custom programs using the RSA Authentication Manager application programming interfaces (APIs). Includes an overview of the APIs and Javadoc for Java APIs.

RADIUS Reference Guide. Describes the usage and settings for the initialization files, dictionary files, and configuration files used by RSA RADIUS.

Authentication Manager Help. Describes day-to-day administration tasks performed in the RSA Security Console. To view Help, click the **Help** tab on the Security Console.

Operations Console Help. Describes configuration and setup tasks performed in the RSA Operations Console. To log on to the Operations Console, see "Logging On to the RSA Operations Console" in the *Administrator's Guide*.

Self-Service Console Frequently Asked Questions. Provides answers to frequently asked questions about the RSA Self-Service Console, RSA SecurID two-factor authentication, and RSA SecurID tokens. To view the FAQ, on the **Help** tab on the Self-Service Console, click **Frequently Asked Questions**.

Tutorials

The following interactive tutorials are included on the RSA Authentication Manager 7.1 DVD or in the download kit:

ConsoleAdministration. Provides Overview and How-To information about the tasks you can perform on the RSA Security Console. You can also access this tutorial from the Security Console by clicking **Help > Console Tutorial**.

SecurIDToken_HowTo. Describes the steps to authenticate using various RSA SecurID tokens. This tutorial can be provided to end users as a training tool.

To view these tutorials, you must have Adobe Flash Player 8 or later. To download the viewer, go to **http://www.adobe.com/products/flashplayer/**.

Getting Support and Service

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
Customer Support Information	**www.rsa.com/support**
RSA Secured Partner Solutions Directory	**www.rsa.com/rsasecured/**

RSA SecurCare Online offers a knowledgebase that contains answers to common questions and solutions to known problems. It also offers information on new releases, important technical news, and software downloads.

The RSA Secured Partner Solutions Directory provides information about third-party hardware and software products that have been certified to work with RSA products. The directory includes Implementation Guides with step-by-step instructions and other information about interoperation of RSA products with these third-party products.

Before You Call Customer Support

Make sure you have access to the computer running the RSA Authentication Manager software.

Please have the following information available when you call:

❑ Your RSA License ID. You can find this number on your license distribution media, or in the RSA Security Console by clicking **Setup** > **Licenses** > **Manage Existing**, and then clicking **View Installed Licenses**.

❑ The Authentication Manager software version number. You can find this in the RSA Security Console by clicking **Help > About RSA Security Console > See Software Version Information**.

❑ The names and versions of the third-party software products that support the Authentication Manager feature on which you are requesting support (operating system, data store, web server, and browser).

❑ The make and model of the machine on which the problem occurs.

1 radius.ini File

The **radius.ini** initialization file is the main configuration file that determines the operation of RSA RADIUS. It contains information that controls a variety of RADIUS functions and operations.

Note: Some parameters or sections may appear in the configuration file that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: If you edit settings in the **radius.ini** file, you must adhere to standard .ini syntax. After making any changes to the **radius.ini** file, you must restart the RADIUS server for the changes to take effect.

[Addresses] Section

By default, the RSA RADIUS server tries to autoconfigure all IPv4 addresses that are reported by name services for the primary host name of the server on which RSA RADIUS is running, so that it can listen for incoming RADIUS packets on all available network interfaces. If IPv6 is enabled, RSA RADIUS autoconfigures its IPv6 addresses and then listens on all interfaces using IPv6 addresses.

You must explicitly configure the IP addresses that you want RSA RADIUS to use in the [Addresses] section of **radius.ini** if RSA RADIUS is running on a multihomed (more than one network interface) server and if any of the following statements apply to your network:

- One or more network interfaces on the server are connected to networks that should not carry RADIUS traffic.
- The server has more than one host name, and IP addresses exist for names other than the primary host name.
- The server has private IP addresses that are not published by name services.

Specifying IPv4 or IPv6 addresses causes the server to listen on only those addresses and ignore all other addresses.

Specifying AutoConfigureIPv4 or AutoConfigureIPv6 causes RSA RADIUS to attempt to discover and configure all IPv4 or IPv6 addresses that belong to the local host automatically.

Example 1

The following example configures RSA RADIUS to listen for RADIUS authentication and accounting requests on the IPv4 address 192.168.12.35 and on all local IPv6 interfaces. Note that IPv6 functionality must be enabled (by setting Enable to 1 in the [IPv6] section of **radius.ini**) before IPv6 addresses can be used.

```
[Addresses]
192.168.12.35
AutoConfigureIPv6
```

To route all of your proxy traffic through a single interface, set the value for ProxySource in the [Configuration] section of **radius.ini** to the appropriate IP address or addresses, which must be listed in the [Addresses] section.

Example 2

The following example routes all proxy traffic through the interface at 192.10.20.30.

```
[Addresses]
192.10.20.30
192.10.20.31

[Configuration]
ProxySource = 192.10.20.30
```

[AuditLog] Section

The [AuditLog] section specifies whether RSA RADIUS maintains an audit log file (*yyyymmdd*.auditlog) to record administrator activities and CCM events. Audit log records are stored in XML format.

Administrator activities include the following:

- Logging on and off by RSA RADIUS administrators
- Creating, modifying, and deleting RSA RADIUS objects (RADIUS clients, users, profiles, proxy targets, proxy realms, tunnels, administrators, authentication policies, or CCM nodes)
- Importing files

CCM events include publication, notification, and download of CCM files.

Note: The audit log does not track changes made through the LDAP configuration interface (LCI).

```
[AuditLog]
;Enable = 0
;LogfilePermissions = owner:group mode
;DaysToKeep = 30
```

The following table lists the [AuditLog] parameters and their functions.

Parameter	Function
Enable	• If set to 0, audit logging is disabled. • If set to 1, audit logging is enabled. Default value is 0.
LogfilePermissions (Solaris/Linux only)	Specifies the owner and access permission setting for the audit log (*yyyymmdd*.auditlog) file. Enter a value for the LogFilePermissions setting in *owner:group permissions* format, where: • *owner* specifies the owner of the file in text or numeric format. • *group* specifies the group setting for the file in text or numeric format. • *permissions* specifies what privileges can be exercised by Owner/Group/Other with respect to the file in text or numeric format. For example, ralphw:1007 rw-r- - - - - specifies that the file owner (ralphw) can read and edit the audit log file, members of group 1007 can read (but not edit) the audit log file, and other users cannot access the audit log file.
DaysToKeep	Specifies the number of days the RSA RADIUS server should retain each authentication acceptance report. Default value is 30 days.

[Configuration] Section

The [Configuration] section of **radius.ini** contains parameters that control basic behavior of RSA RADIUS.

The following table lists the [Configuration] parameters and their functions.

Parameter	Function
AddDestIPAddressAttrToRequest	• If set to 0, RSA RADIUS does not add destination address information to RADIUS requests. • If set to 1, RSA RADIUS adds a Funk-Dest-IP-Address attribute identifying the IP address to which the RADIUS request was sent to the attributes in the packet. All processing that could be performed on an attribute included in the request packet, such as checklist processing, can be performed on this attribute. Default value is 0.

Parameter	Function
AddDestUDPPortAttrToRequest	• If set to 0, RSA RADIUS does not add destination port information to RADIUS requests. • If set to 1, RSA RADIUS adds a Funk-Dest-UDP-Port attribute identifying the UDP port to which the RADIUS request was sent to the attributes in the packet. All processing that could be performed on an attribute included in the request packet, such as checklist processing, can be performed on this attribute. Default value is 0.
AddSourceIPAddressAttrToRequest	• If set to 0, RSA RADIUS does not add source address information to RADIUS requests. • If set to 1, RSA RADIUS adds a Funk-Source-IP-Address attribute identifying the IP address from which the RADIUS request was received to the attributes in the packet. All processing that could be performed on an attribute included in the request packet, such as checklist processing, can be performed on this attribute. Default value is 0.
Apply-Login-Limits	• If set to yes, the maximum number of concurrent connections for each user is enforced, and connection attempts above the limit are rejected. • If set to no, connections above the limit are allowed, but an event is noted in the server log file. Default value is yes.
AuthenticateOnly	• If set to 1, no response attributes are included in the response packet to an AuthenticateOnly (Service-Type 8) request. • If set to 0, the normal response attributes are included in the response. Default value is 1.

Parameter	Function
CheckMessageAuthenticator	Specifies whether validation of Message-Authenticator occurs on receipt of an Access-Request from a network access device or on receipt of an Access-Accept, Access-Reject, or Access-Challenge from a proxy (extended proxy only). • If set to 0, the validation of received Message-Authenticator attributes is disabled. • If set to 1, the validation of received Message-Authenticator attributes is enabled. Default value is 0. **Note:** Validation does not occur for ordinary proxy.
ClassAttributeStyle	• If set to 1, RSA RADIUS uses unencrypted Class attributes with multiple ASCII keys in Access-Accept packets. • If set to 2, RSA RADIUS uses enhanced/encrypted Class attributes in Access-Accept packets. Default value is 2. **Note:** The ClassAttributeStyle parameter must be set to a value of 2 before you can use attribute embedding. For information on attribute embedding, see "[Debug] Section" on page 17.

Parameter	Function
DisableSecondaryMakeModelSelection	• If set to 1, RSA RADIUS looks up the network access device entry by using the source address of the request and sets the make/model according to the information specified for the client. • If set to 0, RSA RADIUS: 1. Looks up the network access device entry by using the source address of the request and sets the make/model according to the information specified for the client. 2. Uses the NAS-IP-Address attribute (if present) to look up the network access device entry. If the IP address is found, override the make/model information identified in Step 1. 3. Uses the NAS-Identifier attribute (if present) to look up the network access device by name. If the name is found, override the make/model information defined in Step 1 or Step 2. Default value is 0.
EnhancedDiagnosticLogging	• If set to no, standard diagnostic logging messages are written to the RADIUS log file when the log level is set to 0. • If set to yes, messages relating to proxy retries, proxy timeouts, and LDAP timeouts, as well as standard diagnostic logging messages, are written to the RADIUS log file (*yyyymmdd*.log) when the log level is set to 0. Default value is no.
FramedIPAddressHint	If set to yes, the attribute Framed-IP-Address is treated as a hint. If this attribute appears in the Access-Request and the user's return list is configured to allocate Framed-IP-Address from a pool, the IP address in the Access-Request is returned instead of a newly-allocated IP address. Default value is no.
LogAccept	• If set to 1, specifies that messages associated with Accepts that meet the current LogLevel should be recorded in the server log file. • If set to 0, messages associated with Accepts are ignored. Default value is 1.

Parameter	Function
LogFilePermissions (Solaris/Linux only)	Specifies the owner and access permission setting for the system log (*yyyymmdd*.log) file. Enter a value for the LogFilePermissions setting in *owner:group permissions* format, where: • *owner* specifies the owner of the file in text or numeric format. • *group* specifies the group setting for the file in text or numeric format. • *permissions* specifies what privileges can be exercised by Owner/Group/Other with respect to the file in text or numeric format. For example, ralphw:1007 rw-r- - - - - specifies that the file owner (ralphw) can read and edit the log file, members of group 1007 can read (but not edit) the log file, and other users cannot access the log file.
LogfileMaxMBytes	• If set to 0 (or if setting is absent), the server log file size is ignored and log file names are date-stamped to identify when they were opened (*YYYYMMDD*.log). • If set to a value in the range 1–2047, the current server log file is closed when it reaches the specified number of megabytes (1024 x 1024 bytes), and a new server log file using the date and time it was opened as its filename (*YYYYMMDD_HHMM*.log) is opened. Default value is 0. **Note:** The size of the log file is checked once per minute. The log file might exceed the size specified in LogFileMaxMBytes, since it does not roll over until the next log size check occurs.
LogHighResolutionTime	• If set to no, the timestamp for entries in the RSA RADIUS log file (*yyyymmdd*.log) are recorded as *hh:mm:ss* (hours:minutes:seconds). • If set to yes, the timestamp for entries in the RSA RADIUS log file (*yyyymmdd*.log) are recorded as *hh:mm:ss:xxx*, where *xxx* represents the number of elapsed milliseconds since the *ss* value changed. Default value is no.

Parameter	Function
LogLevel	Sets the rate at which RSA RADIUS writes entries to the server log file (.LOG): • 0 – Production logging level • 1 – Informational logging level • 2 – Debug logging level Default value is 0.
LogReject	• If set to 0, messages associated with Rejects are ignored. • If set to 1, messages associated with Rejects that meet the current LogLevel should be recorded in the server log file. Default value is 1.
NoNullTermination	• If set to 0, RADIUS reply attributes of type string are sent with a null character at the end of the string (null terminated string). • If set to 1, RADIUS reply attributes of type string are sent without the null character at the end of the string. Entering a value of 1 for this setting is the equivalent of changing all reply attributes of type string to type stringnz. Default value is 0. **Note:** After you change this setting, you must delete the **saved-dicts.bin** file and restart the RSA RADIUS service.
PhantomTimeout	The maximum number of seconds that a phantom session record remains active. As soon as the corresponding accounting start packet is received, a phantom record is discarded. If a phantom record still exists at the end of its timeout period, it is discarded and all resources associated with it are released.
PrivateDir	Name of the location of the RSA RADIUS directory, which contains the database and dictionary files. Default value is the directory in which the RSA RADIUS service/daemon resides.

Parameter	Function
ProxySource	Specifies the IP address of the interface through which all outgoing proxy traffic is routed. The IP address specified for ProxySource must be listed in the [Addresses] section of **radius.ini**. If a ProxySource address is not specified and per-realm control of proxy interfaces is not enabled, RSA RADIUS uses the first interface it finds on the server.
ProxyStripRealm	• If set to 1, the proxy realm decoration is stripped before sending the request downstream. • If set to 0, no realm name stripping is performed. Default value is 1.
SendOnlyOneClassAttribute	When a user's identity information is encrypted during authentication, RSA RADIUS uses a special Class attribute to pass the user's encrypted identity to an accounting server. Because this typically requires more than one Class attribute to be included in the Accept response, and because some Access Points do not support echoing more than one Class attribute, you can use the SendOnlyOneClassAttribute parameter to specify how RSA RADIUS should forward encrypted user identity information. • If set to 1, RSA RADIUS creates a Class attribute containing a Class attribute flag, a server identifier, and a transaction identifier. The user identification data that would normally be stored in the Class attribute(s) is stored in the current sessions table. When RSA RADIUS receives an accounting request, it looks up the Class information in the current sessions table and uses it as if it had arrived in the accounting request packet.

Parameter	Function
SendOnlyOneClassAttribute (continued)	- If set to 0, RSA RADIUS creates one or more Class attributes to return a user's encrypted identity to the Access Point, with the assumption that the AP forwards the Class attribute(s) containing the encrypted user identification information to the accounting server. Default value is 0. **Note:** This feature works only if accounting requests go to the same server that performs authentication. Accounting requests that go to servers other than the authenticating server fail.
StartupTimeout	Specifies the number of seconds RSA RADIUS waits for its startup sequence to finish before timing out. Default value is 360 seconds.
TraceLevel	Specifies the RADIUS packet tracing level: - 0 – No packet tracing - 1 – Parsed content of packets is logged - 2 – Raw content and parsed content of the packet is logged Default value is 0. **Note:** Packet traces are written to the server log file and can be a useful tool for troubleshooting interoperability problems.
TreatAddressPoolsAsDisjoint	- If set to 1, RSA RADIUS treats each IP address pool as though it operates off its own disjoint address space. This disables the normal checks to ensure that an IP address is allocated only to a single address pool. - If set to 0, a single IP address can be allocated only to a single session and from a single IP address pool. Default value is 0. **Note:** To track allocated resources, RSA RADIUS uses the Class attribute to track IP addresses. This attribute contains the IP pool name and IP address.

Parameter	Function
UseNewAttributeMerge	• If set to 1, the new profile and user attribute merging calculation is performed. • If set to 0, the older calculation technique is used. Default value is 1.

[CurrentSessions] Section

The [CurrentSessions] section of **radius.ini** controls the Current Sessions Table.

```
 [CurrentSessions]
;CaseSensitiveUsernameCompare = 1
```

The following table lists the [CurrentSessions] parameters and their functions.

Parameter	Function
CaseSensitiveUsernameCompare	• If set to 1, when the server searches its Current Sessions Table for sessions that have the same user name, it uses case-sensitive lookups. • If set to 0, the server ignores case. Default value is 1.

[Debug] Section

The [Debug] section of **radius.ini** helps debug problems with RSA RADIUS operations by incorporating thread identifiers in log messages. Thread identifiers help you parse the diagnostic log when messages about different RADIUS requests are interleaved.

The syntax for including thread identifiers in diagnostic log messages is:

```
 [Debug]
Log-Thread-ID = yes
```

The following table lists the [Debug] parameters and their functions.

Parameter	Function
Log-Thread-ID	• If set to yes, thread identifiers are included in RSA RADIUS log messages. • If set to no, thread identifiers are omitted from RSA RADIUS log messages. Default value is no.

[EmbedInClass] Section

The [EmbedInClass] section of **radius.ini** identifies attributes that are available during authentication processing which must be made available in accounting requests. Attribute embedding allows billing information to be embedded in a Class attribute returned to RSA RADIUS by a network access device. When RSA RADIUS receives an embedded attribute, it decodes the attribute and places it in the Accounting request according to the settings specified in the **classmap.ini** file (described on page 47).

Note: The ClassAttributeStyle parameter in the [Configuration] section of **radius.ini** must be set to a value of 2 before you can use attribute embedding.

The syntax for embedding attributes is as follows:

```
[EmbedInClass]
responseAttribute={ Clear | Encrypt }[,Remove]
```

The following table lists the [EmbedInClass] parameters and their functions.

Parameter	Function
responseAttribute	Identifies the response attribute to be embedded in the RADIUS Class attribute.
Clear	Specifies that the retrieved information is included in the Class attribute in cleartext format.
Encrypt	Specifies that the retrieved information is encrypted before it is included in the Class attribute.
Remove	Optional parameter that removes the embedded attribute from the Accept-Response packet.

[FailedAuthOriginStats] Section (Windows only)

The [FailedAuthOriginStats] section of **radius.ini** enables you to identify when a specific network access device is associated with certain Windows Performance Monitor (perfmon) counters. This helps you to identify a specific region of your network that might be having difficulties. The syntax is as follows:

```
[FailedAuthOriginStats]
RADIUSclient=IDnumber
RADIUSclient=IDnumber
M
```

where

- *RADIUSclient* is the name of a network access device as defined in the RADIUS Clients window
- *IDnumber* is a number in the range 1 to 16.

These numbers map to the following RSA RADIUS perfmon counters:

```
Failed Auths - 1
Failed Auths - 2
M
Failed Auths - 16
```

For example, if you map a RADIUS client named herman to the number 3, then the perfmon counter Failed Auths - 3 tells you the number of failed authentication requests that have originated from RADIUS client herman.

```
[FailedAuthOriginStats]
herman=3
```

[HiddenEAPIdentity] Section

The [HiddenEAPIdentity] section of **radius.ini** allows the known inner identity of EAP/TTLS and EAP/SIM protocols to be included in the Access-Accept message returned in response to an authentication request.

The syntax is as follows:

```
[HiddenEAPIdentity]
IncludeInAcceptResponse=0|1
ResponseAttribute = attributeName[, replaceAttribute]
```

The following table lists the [HiddenEAPIdentity] parameters and their functions.

Parameter	Function
IncludeInAcceptResponse	• If set to 0, inclusion of the inner identity in Access-Accept responses is disabled. • If set to 1, RSA RADIUS includes the inner identity in the specified attribute of an Access-Accept response. Default value is 0.
attributeName	Identifies the attribute in which to include the inner identity in an Access-Accept message. If this value is omitted, the User-Name attribute is used. The attributeName value can be any string attribute, including a VSA, that is defined in an attribute dictionary.
[, replaceAttribute]	Identifies the Access-Accept attribute that retains the original value of the attribute specified in the *attributeName* argument. If a replacement value is not specified, the value of the original attribute is lost.

[IPv6] Section

```
[IPv6]

   Enable = 0
   DynamicNameResolution = 2
   IPv6LinkLocalUnicastScopeId = 0
   IPv6SiteLocalUnicastScopeId = 0
```

The [IPv6] section of **radius.ini** controls IPv6 network transport features.

The following table lists the [IPv6] parameters and their functions.

Parameter	Function
Enable	Determines whether IPv6 networking is enabled in RSA RADIUS. • If set to 0, IPv6 networking is disabled, and other values in the IPv6 section of **radius.ini** are ignored. • If set to 1, IPv6 networking is enabled. Default value is 0. **Note:** IPv4 networking is always enabled in RSA RADIUS.
DynamicNameResolution	Determines whether the RSA RADIUS server tries to use IPv6 name services (DNSv6) to resolve host names. • 0 – Do not use IPv6 name services. IPv4 name services are not affected by this setting. • 1 – Use only IPv6 name services. IPv4 name services are disabled by this setting. • 2 – Use IPv6 name services first; use IPv4 name services in case of failure. Default value is 2.
IPv6LinkLocalUnicastScopeId	Specifies an interface name (such as hme0) or index (4) for Solaris/Linux hosts. Specifies an interface index (4) for Windows hosts. If set to 0, RSA RADIUS does not use link local addresses. Default value is 0.
IPv6SiteLocalUnicastScopeId	• Solaris/Linux: Specifies an interface name (such as hme0) or index (4). • Windows: Specifies an interface index (4). If set to 0, RSA RADIUS selects the site local scope ID automatically. Default value is 0.

[MsChapNameStripping] Section

The [MsChapNameStripping] section of **radius.ini** specifies whether you want RSA RADIUS to try to strip domain information from user names when it tries to match its user entry to the user name/password hash forwarded by the enduser. This feature is useful in situations where the user name in the RSA RADIUS database includes characters the enduser host considers domain information, which it deletes before computing its hash of the user's credentials.

If this feature is enabled:

1. RSA RADIUS scans the user name in its database looking for delimiter characters that might indicate a domain is prefixed to the user name. If a prefix delimiter character is found, the server strips that character (and all characters to the left of the delimiter), generates its own hash of the user's credentials, and compares the result to the hashed credentials forwarded by the enduser to determine if a match is found.
2. If a prefix delimiter is not found (or if the hashed credentials do not match after the prefix is stripped), RSA RADIUS scans the user name looking for delimiter characters that might indicate a domain is suffixed to the user name. If a suffix delimiter character is found, the server strips that character (and all characters to the right of the delimiter), generates its own hash of the user's credentials, and compares the result to the hashed credentials forwarded by the enduser to determine if a match is found.
3. If neither a prefix delimiter nor a suffix delimiter is found (or if a delimiter was found but the hashed credentials did not match), the server uses the entire username string to generate the hashed credentials and compares the result to the hashed credentials forwarded by the enduser to determine if a match is found.

The syntax for the [MsChapNameStripping] section is as follows:

```
[MsChapNameStripping]
Enable=1
Prefix=\\
Suffix=/@
```

The following table lists the [MsChapNameStripping] parameters and their functions.

Parameter	Function
Enable	• If set to 0 (or omitted), MS-CHAP name stripping is disabled. • If set to 1, MS-CHAP name stripping is enabled. Default value is 0.
Prefix	A list of as many as five ASCII characters to strip from the prefix. If a space character appears in the list, the entire list must be surrounded by quotation marks. Enter a double backslash (\\) to indicate that you want to strip the backslash character. A double backslash counts as one character in the list. Default value is \\.

Parameter	Function
Suffix	A list of as many as five ASCII characters to strip from the suffix. If a space character appears in the list, the entire list must be surrounded by quotation marks. Enter a double backslash (\\) to indicate you want to strip the backslash character. A double backslash counts as one character in the list. Default value is /@.

[Ports] Section

The [Ports] section of **radius.ini** provides a method for setting the UDP ports used by RSA RADIUS:

- If one or more UDPAuthPort settings are specified in the [Ports] section of **radius.ini**, the port numbers in this section are the only ones on which the server listens for authentication requests. Similarly, if one or more UDPAcctPort settings are specified, they are the only ones on which the server listens for accounting requests.

  You can specify as many as 64 port numbers on a Windows server and as many as 4096 ports on a Solaris or Linux server. If this limit is exceeded, the RADIUS authentication subcomponent fails to initialize.
- If no UDPAuthPort or UDPAcctPort settings are present in the [Ports] section, the server attempts to read the port numbers associated with radius service (authentication) and radacct (accounting) in **/etc/services**. If successful, the server listens on these port numbers. No more than one port can be specified for the radius service or for the radacct service.
- If no UDPAuthPort settings are present in the [Ports] section and no radius service or radacct is listed in the **/etc/services** file, the server listens for authentication requests on UDP ports 1645 and 1812 for authentication and UDP ports 1646 and 1813 for accounting.

Note: Any failure to bind to one of the selected UDP ports causes the affected subcomponent (authentication or accounting) to fail to initialize.

If the server will function as a proxy forwarding server, you can specify a block of UDP port numbers from which the proxy RADIUS ports are allocated. Proxy RADIUS allocates port numbers in sets of eight. Port numbers in an allocated block do not have to be contiguous: if a UDP port number that falls in the proxy RADIUS range is in use, proxy RADIUS skips over it.

The following table lists the [Ports] parameters and their functions.

Parameter	Function
SecureTcpAdminAddress	Specifies the IP address of the administrative interface used for communication between the administration client and the RSA RADIUS server. If not specified, any network interface on the RSA RADIUS server accepts a connection from the administration client.
SecureTcpAdminPort	Specifies the TCP port used for communication between the administration client and the RSA RADIUS server. Default value is 1813.
TCPControlAddress	Specifies the IP address of the administrative interface on the RSA RADIUS server used for SNMP and CCM/ replication communication. If not specified, any network interface on the RSA RADIUS server can be used for SNMP and CCM traffic.
TCPControlPort	Specifies the TCP port used for SNMP and CCM/replication communication. Default value is 1812.
UDPAuthPort	Specifies the UDP port(s) used for authentication. If you use more than one port, specify each port number on a separate line. Default values are 1645 and 1812.
UDPAcctPort	Specifies the UDP port(s) used for accounting. If you use more than one port, specify each port number on a separate line. Default values are 1646 and 1813.
UDPProxyPortBlockLength	Specifies the number of addresses in the port number range used for proxy RADIUS communication. Default value is 64.
UDPProxyPortBlockStart	Specifies the starting port number in the port number range used for proxy RADIUS communication. Default value is 28000. **Note:** If you change the default value, choose a number range that does not overlap with well-known UDP ports and proprietary UDP ports on your network. **Note:** You might need to configure network firewalls to allow ports in the specified number range to pass.

For example:

```
 [Ports]
SecureTcpAdminPort = 1813
SecureTcpAdminAddress = 192.168.12.15
TcpControlPort = 1812
TCPControlAddress = 192.168.15.55
UDPAuthPort = 1645
UDPAuthPort = 1812
UDPAcctPort = 1646
UDPAcctPort = 1813
UDPProxyPortBlockStart = 28000
UDPProxyPortBlockLength = 64
```

The UDP port assignments entered in the [Ports] section of the **radius.ini** file override the UDP port assignments specified in the **/etc/services** file. For more information, see Chapter 3, "services File."

[SecurID] Section

The [SecurID] section of **radius.ini** contains items specific to RSA SecurID authentication for ISDN users. It provides information that allows RSA RADIUS to cache the user's credentials temporarily after a successful SecurID authentication. This technique is necessary to permit a second ISDN B-channel to be authenticated during the user's session. RSA RADIUS uses the cached token to authenticate the second channel.

Note: If this feature is not enabled, users who want to authenticate against a SecurID database through an ISDN connection that "bonds" both B-channels will fail to authenticate due to a SecurID security violation. ISDN users running only one B-channel are not affected.

The following table lists the [SecurID] parameters and their functions.

Parameter	Function
CachePasscodes	• If set to yes, RSA SecurID passcode caching is enabled. • If set to no, RSA SecurID passcode caching is disabled. Default value is no.
SecondsToCachePasscodes	The number of seconds to retain the cached SecurID passcode (PIN and token code). Default value is 60 seconds.

[UserNameTransform] Section

The [UserNameTransform] section lets you specify a rule for transforming user names in RADIUS requests from the form in which they are received to a form in which they can be processed. This can be useful when the form in which users supply their names to the network access device is not compatible with the form in which the RADIUS server applies its rules for proxy forwarding or with the form that the authentication system requires.

The user name transformation rule used to convert input strings to output strings is based on an input format and an output format. The user name transformation rule is applied to user names appearing in RADIUS requests. The user name from the RADIUS request is parsed based on the input format.

- If the user name does not conform to the input format, the rule does not apply and the user name is unchanged.
- If the rule does apply, the parsed elements of the user name are formatted based on the output format to construct the transformed user name:
  - The User-Name from the Access-Accept (or Acct-Start/Acct-Stop) is compared to the input format rule.
  - If the User-Name matches the rule, it is modified into the output format, and authentication continues.
  - If the User-Name does not match the input format, no modification occurs, and authentication continues.

The transformed user name replaces the original user name in RADIUS processing, just as if the transformed user name had been included in the request. The decision to proxy-forward the packet is based on the transformed user name, and all authentications are based on the transformed user name.

Format strings can be any sequence of characters, and can contain embedded variables enclosed in angle brackets (< >). The backslash (\) is an escape character within text, used to represent literal characters. Within variable names, a backslash is treated as a character, not as an escape; and therefore, variable names may not include right angle brackets (>).

The literal text should be composed of characters not expected to be found in the variable elements. Use punctuation characters such as a slash (/) or an at-sign (@), rather than letters or numbers.

The user name transformation rule can be applied to authentication packets, accounting packets, or both.

```
[UserNameTransform]
In=<input format>
Out=<output format>
Authentication=< yes | no >
Accounting=< yes | no >
```

The following table lists the [UserNameTransform] parameters and their functions.

Parameter	Function
In	A format string identifying the input format for user names. For example, <user>@<realm>.
Out	A format string identifying the output format for user names. For example, <user>.
Authentication	Set to Yes to enable the transform for authentication requests. Default value is Yes.

Parameter	Function
Accounting	Set to Yes to enable the transform for accounting requests. Default value is Yes.
Proxy	Set to Yes to enable the transform for proxied requests. Default value is Yes.

For example, the following settings transforms george@acme.com to george:

```
In = <user>@<realm>
Out = <user>
```

The following settings transform abc/martha@bigco.com to bigco.com::abc/martha:

```
In = <prefix>/<user>@<realm>
Out = <realm>::<prefix>/user
```

2 sbrd.conf File (Solaris/Linux only)

The **sbrd.conf** file is an executable Bourne shell script that is invoked by the **sbrd** process to initialize the execution environment for RSA RADIUS.

Note: Some parameters or sections may appear in the configuration file that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: Do not modify the sbrd script. Instead, modify **sbrd.conf** to have appropriate operational configurations, such as changing file mask, ulimit, and watchdog.

For example:

```
#!/bin/sh
##############
# sbrd.conf
##############
ULIMIT_CORE_SIZE=""
ULIMIT_CORE_COUNT=3
ULIMIT_OPEN_FILES=1024

RADIUSUMASK=""
RADIUS_HIGH_FDS=1
ORACLE_MSB_FILE=”ORACLE_HOME/rdbms/mesg/ocius.msb”

# Radius executable, options, and arguments
RADIUS="radius"
RADIUSOPTS=""
RADIUSARGS="sbr.xml"
RADIUS_PRIVATE_DIR="$RADIUSDIR"

# Watchdog executable, options, and arguments
WATCHDOGENABLE=0
WATCHDOG="radiusd"
WATCHDOGOPTS="--config $RADIUSDIR/radiusd.conf --pidfile
$RADIUSDIR/radius.pid"
WATCHDOGARGS="$RADIUSDIR/$SELF"
```

Note: Do not include spaces in parameter settings in the **sbrd.conf** file.

Correct: ULIMIT_CORE_COUNT=3
Incorrect: ULIMIT_CORE_COUNT = 3

The following table lists the **sbrd.conf** parameters and their functions.

Parameter	Function
ULIMIT_CORE_SIZE	Specifies the size of core files generated if RSA RADIUS fails. • If set to a value, ULIMIT_CORE_SIZE specifies the maximum size for core files in 512-byte blocks (Solaris) or in 1024-byte blocks (Linux). • If set to disabled, RSA RADIUS uses the current environment without changes. • If set to ““ (two double-quotes with no space between), RSA RADIUS uses the current environment, making adjustments as needed. Default value is ““.
ULIMIT_CORE_COUNT	Specifies the number of core files maintained on the RSA RADIUS server. If the maximum number of core files already exists on the server, RSA RADIUS discards the oldest core files and generates a new core file if it fails. • If set to a number in the range 0–999,999,999, the server maintains the specified number of core files. • If set to unlimited, RSA RADIUS does not discard existing core files if it generates a new one. • If set to disabled, RSA RADIUS uses the current environment without changes. • If set to ““ (two double-quotes with no space between), RSA RADIUS uses the current environment, making adjustments as needed. Default value is 3.
ULIMIT_OPEN_FILES	Specifies the number of open files that the RSA RADIUS process can have open at one time. • If set to a number in the range 256–1024, the server maintains the specified number of open files. • If set to disabled, RSA RADIUS uses the current environment without changes. • If set to ““ (two double-quotes with no space between), RSA RADIUS uses the current environment, making adjustments as needed. Default value is 1024.

Parameter	Function
RADIUSMASK	Specifies the file permissions that are withheld when new log files are created. • If set to a umask argument, log files are created with the specified permissions withheld from Owner, Group, and Other users. • If set to "", log files are created with the default access permissions established by the ambient umask for Owner, Group, and Other users. For information on how to configure and use umask to control file permission settings, see the chapter "Managing RSA RADIUS" in the *Administrator's Guide*.
RADIUS_HIGH_FDS	• If set to 0, management of file descriptors is disabled. You can set RADIUS_HIGH_FDS to 0 if you specified a value of 256 or lower for the ULIMIT_OPEN_FILES parameter. • If set to 1, management of file descriptors is enabled. Set RADIUS_HIGH_FDS to 1 if you specified a value greater than 256 for the ULIMIT_OPEN_FILES parameter. Default value is 1.
ORACLE_MSB_FILE	Specifies the absolute path to the locale-specific Oracle message file. • If you enter the path name to a message file, the file descriptor that is returned is greater than 255. • If you enter "", RSA RADIUS uses the descriptor returned by a standard library open() call.
RADIUS	Default value is "radius". Do not change this value unless instructed to do so by technical support.
RADIUSOPTS	Specifies options used when running RSA RADIUS. Default value is "". Do not change this value unless instructed to do so by technical support.
RADIUSARGS	Default value is "sbr.xml". Do not change this value unless instructed to do so by technical support.
RADIUS_PRIVATE_DIR	Default value is "$RADIUSDIR". Do not change this value unless instructed to do so by technical support.

Parameter	Function
WATCHDOGENABLE	• If set to 0, the RSA RADIUS watchdog process, which restarts RSA RADIUS if it fails, is disabled. • If set to 1, the RSA RADIUS watchdog is enabled. Default value is 0.
WATCHDOG	Specifies the name of the RSA RADIUS watchdog process. Default value is radiusd. Do not change this value unless instructed to do so by technical support.
WATCHDOGOPTS	Default value is --config $RADIUSDIR/radiusd.conf --pidfile $RADIUSDIR/radius.pid. Do not change this value unless instructed to do so by technical support.
WATCHDOGARGS	Default value is $RADIUSDIR/$SELF. Do not change this value unless instructed to do so by technical support.

3 services File

The **services** file can be used to assign default UDP ports for RADIUS communications to and from the RSA RADIUS server. RSA RADIUS reads the **services** file at startup. Among the items of information in the **services** file are the port assignments for RADIUS authentication and accounting services.

Note: Some parameters or sections may appear in the configuration file that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: If you edit settings in the **services** file, you must adhere to the standard syntax. After making any changes to the **services** file, you must restart the service that you modified for the changes to take effect.

The following figure illustrates part of a sample services file.

```
# This file contains port numbers for well-known services
# defined by IANA. Format:
#
# <service> <port number>/<protocol> [aliases...]
[#<comment>]
#

echo                 7/tcp
echo                 7/udp
discard              9/tcp    sink null
discard              9/udp    sink null
systat              11/tcp    users       #Active users
systat              11/tcp    users       #Active users
```

The location of the **services** file depends on your operating system:

- Solaris/Linux: **/etc/** (may be mapped using NIS or NIS+)
- Windows: **C:\WINDOWS\system32\drivers\etc\**

If no entry for radius or radacct is found in the **services** file, RSA RADIUS uses the default UDP ports (1645 and 1812 for authentication, 1646 and 1813 for accounting).

RSA RADIUS can be configured to use any available UDP ports for authentication and accounting:

1. Use a text editor to open the **services** file.
2. To set the port for authentication, set the value of the radius parameter. For example:

```
radius 1812/udp # RADIUS authentication protocol
```

3. To set the port for accounting, set the value of the radacct parameter. For example:

```
radacct 1813/udp # RADIUS accounting protocol
```

Note: Port number assignments made in the **radius.ini** file override the assignments made in this file. For more information, see "[Ports] Section" on page 22.

You can determine the ports that RSA RADIUS is using at any time by examining the server log file for that time period.

Note: If another RADIUS server is running on the same host, you must modify the **services** file to avoid port number conflicts if the other RADIUS server binds to the default ports before RSA RADIUS starts.

4 securid.ini File

The **securid.ini** file lets you replace the default prompt strings used in RSA SecurID authentication with customized strings. Customized prompt strings are useful in situations where authentication is to be performed by means of RSA SecurID and the default prompt strings are too long for the screen on the authentication device.

Note: Some parameters or sections may appear in the initialization file that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: If you edit settings in the **securid.ini** file, you must adhere to standard .ini syntax. After making any changes to the **securid.ini** file, you must restart the RADIUS server for the changes to take effect.

If the **securid.ini** file is present in the RSA RADIUS server directory, RSA RADIUS uses prompt strings specified in the file instead of the default prompt strings. Sets of strings can be substituted in whole or in part. If a string is not represented by an entry in the **securid.ini** file, RSA RADIUS uses the default prompt string.

[Configuration] Section

The [Configuration] section of **securid.ini** specifies RSA SecurID access settings.

```
[Configuration]
Enable = 1
AllowSystemPins = 0
CheckUserAllowedByClient = 1
DefaultProfile = DEFAULT
```

The following table lists the [Configuration] parameters and their functions.

Parameter	Function
Enable	• If set to 1, RSA RADIUS can authenticate users by means of RSA SecurID. • If set to 0, RSA RADIUS cannot authenticate users by means of RSA SecurID. Default value is 1.

Parameter	Function
AllowSystemPins	• If set to 1, users who are configured in the RSA Authentication Manager to receive a system-generated PIN when in New PIN mode are accepted. • If set to 0, users who are configured in the RSA Authentication Manager to receive system-generated PIN when in New PIN mode are rejected. Default value is 0.
CheckUserAllowed ByClient	• If set to 1, the RADIUS server verifies the user is allowed to connect through the network access device. • If set to 0, the RADIUS server does not verify the user is allowed to connect through the network access device. Default value is 1. **Note:** If this parameter is set to 1, RADIUS clients must be configured as Agent Hosts in Authentication Manager. For more information, see “Managing RADIUS Clients” in the chapter “Managing RSA RADIUS” in the *Administrator’s Guide*.
DefaultProfile	Default profile to be assigned to a user if the Authentication Manager does not return a profile. Default value is DEFAULT.

[Server_Settings] Section

The [Server_Settings] section of **securid.ini** specifies settings for Extended One-Time Password (EOTP or EAP-15) and Protected One-Time Password (POTP or EAP-32) authentication.

```
 [Server_Settings]
Greeting =
Return_MPPE_Keys = 1
```

The following table lists the [Server_Settings] parameters and their functions.

Parameter	Function
Greeting	A string of as many as 80 characters returned to a network access device after a user is authenticated. For example, “Welcome to RSA Software.”

Parameter	Function
Return_MPPE_Keys	If set to 0, the module does not forward RADIUS MS-MPPE Send-Key and MS-MPPE-Recv-Key attributes. If set to 1, the 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. This is necessary for the Access Point to key the WEP encryption. If the Access Point is authenticating only end users and WEP is not being used, this attribute may be set to 0. Default value is 1.

[Prompts] Section

If the **securid.ini** file is present in the RSA RADIUS server directory, RSA RADIUS uses prompt strings specified in the file instead of the default prompt strings. Sets of strings can be substituted in whole or in part. If a string is not represented by an entry in the **securid.ini** file, RSA RADIUS uses the default prompt string.

Substitution String Formats

Substitution strings use %s to mark locations at which variable text is to be substituted. Strings can have no %s placeholders, exactly one %s placeholder, or exactly two %s placeholders. When writing your own prompt strings, you must supply strings with the expected number of %s placeholders. String names include a reminder suffix that reflects the number of %s placeholders:

- Strings that require two %s placeholders have names with a _S_S suffix. The first %s placeholder typically presents a number range (“4 to 8”). The second %s placeholder specifies “characters” or “digits” (or the equivalent, as configured in the Characters and Digits settings).
- Strings that require one %s placeholder have names with a _S suffix. The %s placeholder is replaced with a system-generated PIN.
- Strings that do not require %s placeholders have names with no suffix.

If a string in the **securid.ini** file is formatted incorrectly, it is ignored and the default prompt string is used.

The following table lists the formatting conventions for the **securid.ini** file.

Convention	Explanation
\b	Backspace; not typically used
\f	Formfeed
\n	Newline; typically used in conjunction with \r
\r	Carriage return; typically used in conjunction with \n

Convention	Explanation
\t	Horizontal tab
\v	Vertical tab; not typically used
\\	Displayed backslash
\’	Displayed single-quote character
\”	Displayed double-quote character

If other characters in a substitution string are preceded by a backslash, the backslash is ignored and the character is displayed unchanged.

Quoted Strings

Trailing white space is ignored when an unquoted prompt string is read into RSA RADIUS. If you want a substitution string to include trailing white space, insert double-quote marks at the beginning and end of the string, enclosing the white space you want to include. For example, if you want a string to be displayed as the word PIN followed by a colon followed by a single space, you would enter **StringName="PIN: "** (with a space between the colon and the closing double-quote character).

Example 1: Verbose Substitution Strings

The following figure lists the default prompt strings, which may be too long for some SecurID displays. Although text lines in this display appear to wrap to a second line, text wrapping is not supported in **securid.ini** entries.

```
;[Prompts]
;InputNextCode        = \r\nPlease Enter the Next Code from Your Token:
;InputMutChoose_S_S = \r\n  Enter your new PIN, containing %s %s,\r\n
or\r\n   <Ctrl-D> to cancel the New PIN procedure:
;InputCannotChoose    = \r\n   Press <Return> to generate a new PIN and
display it on the screen,\r\n          or\r\n   <Ctrl-D> to cancel the New
PIN procedure:
;InputMayChoose_S_S  = \r\n  Enter your new PIN, containing %s %s,\r\n
or\r\n   Press <Return> to generate a new PIN and display it on the
screen,\r\n           or\r\n   <Ctrl-D> to cancel the New PIN procedure:
;InputReadyForPin     = \r\n\r\nARE YOU PREPARED TO HAVE THE SYSTEM
GENERATE A PIN? (y or n) [n]:
;InputReadyForPin_1_S = \r\n\r\nPIN:         %s\r\n\r\n  10 second display
or Hit RETURN to continue.
;InputReenterPin      = \r\n                 Please re-enter new PIN:
;InputReenterPin_1    = \r\nPINs do not match. Please try again.\r\n
;OutputReject         = \r\n\r\nPIN rejected. Please try
again.\r\n\r\nEnter PASSCODE:
;OutputChange         = \r\n\r\nWait for the code on your card to change,
then log in with the new PIN\r\n\r\nEnter PASSCODE:
;OutputAccepted       = \r\nPASSCODE Accepted\r\n
;OutputDenied         = \r\nAccess Denied\r\n\r\n\r\nEnter PASSCODE:
;OutputNoPassReqd     = \r\nPASSCODE Not Required\r\n
;OutputDeniedFinal    = \r\nAccess Denied\r\n\r\n
;Characters           = characters
;Digits               = digits
```

Example 2: 2 x 40 Display Substitution Strings

The following figure displays prompt strings designed for a 2 line x 40 character display. Although text lines in this display appear to wrap to a second line, text wrapping is not supported in **securid.ini** entries.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
; BEGINNING OF 2 lines by 40 characters prompts, these use the full 40
; character width (not including "\r\n") and one or two lines
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
;[Prompts]
;InputNextCode        = Please Enter the Next Code from\r\nYour Token
;InputMustChoose_S_S  = Enter your new PIN (%s %s)
;InputCannotChoose    = Press <Return> to generate a new PIN and\r\ndisplay
it
;InputMayChoose_S_S   = Enter new PIN (%s %s) or press\r\n<Return> to
generate a new one
;InputReadyForPin     = ARE YOU PREPARED TO HAVE THE SYSTEM\r\nGENERATE A
PIN? (y or n) [n]
;InputReadyForPin_1_S = PIN: %s, 10 second display or\r\npress <Return> to
continue
;InputReenterPin      = Please re-enter new PIN
;InputReenterPin_1    = PINs do not match,\r\nPlease try again
;OutputReject         = PIN Rejected, please try again\r\nEnter PASSCODE
;OutputChange         = Wait for the code on your card to change\r\n then
log in with new PIN, Enter PASSCODE
;OutputAccepted       = PASSCODE Accepted
;OutputDenied         = Access Denied\r\nEnter PASSCODE
;OutputNoPassReqd     = PASSCODE Not Required
;OutputDeniedFinal    = Access Denied
;Characters           = chars
;Digits               = digits
```

Example 3: Terse Substitution Strings

The following figure displays prompt strings designed to be parsed by a program at the client endpoint rather than read by a user.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
; BEGINNING OF extremely terse prompts. These are appropriate for automatic
; interpretation by another program which parses the prompts. A well
trained
; end user could use these.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;
;[Prompts]
;InputNextCode         = Next code
;InputMustChoose_S_S   = Must choose
;InputCannotChoose     = Cannot choose
;InputMayChoose_S_S    = May choose (%s, %s)
;InputReadyForPin      = Ready for pin
;InputReadyForPin_1_S  = Ready for pin 1
;InputReenterPin       = Reenter pin
;InputReenterPin_1     = Reenter pin 1
;OutputReject          = Reject
;OutputChange          = Change
;OutputAccepted        = Accepted
;OutputDenied          = Denied
;OutputNoPassReqd      = No pass reqd
;OutputDeniedFinal     = Denied final
;Characters            = chars
;Digits                = digits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

5 Attribute Processing Files

This chapter describes the usage and settings for the RSA RADIUS attribute processing and dictionary files, which specify RADIUS attributes.

Note: Some parameters or sections may appear in the attribute processing and dictionary files that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: If you edit settings in the .ini files, you must adhere to standard .ini syntax. After making any changes to the .ini files, you must restart the RADIUS server for the changes to take effect.

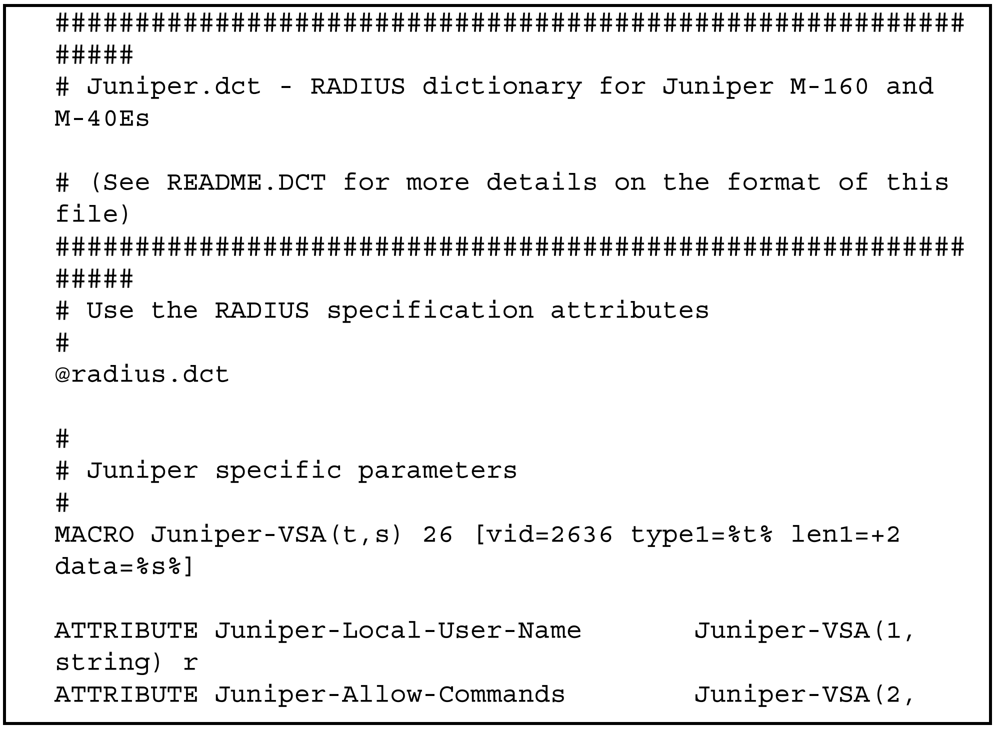
Overview

For each product listed in the **vendor.ini** file, RSA RADIUS provides a dictionary (.dct) file. Dictionary files enable RSA RADIUS to exchange attributes with RADIUS clients. Like initialization files, dictionary files are loaded at startup time, and reside in the RSA RADIUS directory:

```
*.dct
dictiona.dcm
```

- Dictionary files identify the attributes RSA RADIUS should expect when receiving RADIUS requests from a specific type of device.
- Dictionary files identify the attributes RSA RADIUS should include when sending a RADIUS response to a specific type of device.

The following figure illustrates the format of a sample dictionary file.



```
##########################################################
#####
# Juniper.dct - RADIUS dictionary for Juniper M-160 and
M-40Es

# (See README.DCT for more details on the format of this
file)
##########################################################
#####
# Use the RADIUS specification attributes
#
@radius.dct

#
# Juniper specific parameters
#
MACRO Juniper-VSA(t,s) 26 [vid=2636 type1=%t% len1=+2
data=%s%]

ATTRIBUTE Juniper-Local-User-Name       Juniper-VSA(1,
string) r
ATTRIBUTE Juniper-Allow-Commands        Juniper-VSA(2,
```

Dictionary File Location

Windows: Dictionary files must be placed in the same directory as the RSA RADIUS service. While starting up, RSA RADIUS scans its home directory for all files with an extension of .dct (standard dictionary files) and uses the list to create a “master” dictionary, which includes all known attributes.

Solaris/Linux: Dictionary files must be placed in the same directory as the RSA RADIUS daemon. During initialization, RSA RADIUS reads the file **dictiona.dcm** in the server directory to get a list of files with an extension of .dct (standard dictionary files) and uses the list to create a “master” dictionary, which includes all known attributes.

Dictionary File Records

Records in a dictionary file must begin with one of the keywords listed in the following table.

Keyword	Function
@	Include the referenced file
ATTRIBUTE	Define a new attribute
VALUE	Define a named integer value for an attribute
MACRO	Define a macro used to simplify repetitive definitions

Keyword	Function
OPTIONS	Define options beyond the scope of attribute definitions
#	Ignore this text (comment)

Editing Dictionary Files

The product-specific files shipped with RSA RADIUS reflect specific vendors' implementations of RADIUS clients. Therefore, you do not usually need to modify the dictionary files shipped with RSA RADIUS. However, if your network access device vendor provides information about a new product, a new attribute, or a new value for an attribute, you can add this information to your existing RSA RADIUS configuration by editing dictionary files.

Before you edit an existing dictionary file or create a new one, you must do the following to integrate your changes into RSA RADIUS:

1. Add a new vendor-product entry to **vendor.ini** so that you can reference the new dictionary while configuring RSA RADIUS.
2. Place your dictionary file in the same directory as the RSA RADIUS service or daemon.
3. Edit the **dictiona.dcm** file so that it includes your new dictionary file.
4. Stop and restart the server.

Include Records

Records in a dictionary file that begin with the @ character are treated as special include records. The string that follows the @ character identifies the name of a dictionary file whose contents are to be included. For example, the entry @vendorA.dct would include all of the entries in the file **vendorA.dct**.

Include records are honored only one level deep. For example, if file **vendorA.dct** includes file **radbase.dct** and **radbase.dct** includes **radacct.dct**, **vendorA.dct** incorporates records in **radbase.dct** but not those in **radacct.dct**.

Master Dictionary File

The master dictionary **dictiona.dcm** consists of include records that reference vendor-specific dictionaries. The order in which vendor-specific dictionaries are included in the master dictionary has significance only if two vendor-specific dictionaries contain conflicting definitions for the same attribute or attribute value. The first definition of an attribute or attribute value takes precedence over later definitions of the same attribute or attribute value. For example, if master dictionary **dictiona.dcm** consists of the following include records:

```
@vendorA.dct
@vendorB.dct
@vendorC.dct
```

then attributes and attribute values defined in **vendorA.dct** override attributes and attribute values defined in **vendorB.dct** or **vendorC.dct**, and attributes and attribute values in **vendorB.dct** override attributes and values defined in **vendorC.dct**.

ATTRIBUTE Records

Attribute records conform to the following syntax:

```
ATTRIBUTE attrib_name attrib_id syntax_type flags
```

Parameter	Function
attrib_name	Name of the attribute (up to 31 characters with no embedded blanks).
attrib_id	Integer in the range 0 to 255 identifying the attribute's encoded RADIUS identifier.
syntax_type	Syntax type of the attribute.
flags	Defines whether an attribute appears in the checklist, the return list (or both), whether it is multivalued and whether it is orderable.

Note: One limitation of standard dictionary files (the attrib_id of all the attribute records must be unique) is waived for the master dictionary file. Multiple vendors can define different attribute names for the same attribute identifier (assuming the attribute identifier is not already used in the base RADIUS specification). Since attributes in the RSA RADIUS database are stored by name (rather than by attrib_id), this introduces no ambiguity into the database.

The following example illustrates a typical attribute record:

```
ATTRIBUTE Framed-IP-Netmask 9 ipaddr Cr
```

This attribute record specifies all of the following:

- An attribute named Framed-IP-Netmask is supported.
- The attribute’s encoded RADIUS identifier is 9.
- The attribute must use the syntax of an IP address.
- Flag characters specify that the attribute can appear multiple times in a checklist (C) and at most one time in a return list for User or profile entries (r) in the RSA RADIUS database.

Attribute Name and Identifier

No two attribute records in a single dictionary file should have the same attrib_name or attrib_id. If a duplicate attrib_name or attrib_id is encountered, the later definition of the attribute is ignored in favor of the earlier one.

Syntax Type Identifier

Standard syntax_type identifiers are listed in the following table.

Syntax Type	Function
hexadecimal	Hexadecimal string.
hex4	4-byte (32-bit) unsigned hexadecimal number.
int1	1-byte (8-bit) unsigned decimal number.
int4, integer	4-byte (32-bit) unsigned decimal number.
signed-integer	4-byte (32-bit) signed decimal number. A number with a 1 in the first bit position is interpreted as a negative number.
ipaddr	IP address or IP netmask attribute.
ipaddr-pool	IPv4 address selected from an IP address pool.
ipxaddr-pool	IPX network number selected from an IPX address pool.
string	String attribute (includes null terminator).
stringnz	String attribute (without null terminator).
time	Time attribute (number of seconds since 00:00:00 GMT, 1/1/1970).

Note: Signed integer support is limited to attributes received in packets and processing relating to those attributes, such as accounting logs, authentication logs, authentication reports, and SQL plug-ins.

Compound Syntax Types

In addition to the standard syntax_type identifiers listed in the following table, the dictionary can accommodate compound syntax types for use in defining vendor-specific attributes. Instead of a single syntax_type identifier, one or more of the options listed can be combined inside square brackets to form a compound syntax type.

Option	Function
vid=*nnn*	The device manufacturer's SMI Network Management Private Enterprise code (assigned by ISO) in decimal form.
type*N*=*nnn*	Type setting for vendor-specific attribute as defined in the RADIUS specification; *N* specifies the length of the field (in bytes), *nnn* specifies the decimal value of the field.

Option	Function
len*N*=*nnn*	Length field for vendor-specific attribute as defined in the RADIUS specification; *N* specifies the length of the field (in bytes), *nnn* specifies the decimal value of the field (a plus sign prior to the value indicates that the length of the data portion is to be added to *nnn* to obtain the actual length).
data=*syntax_type*	The actual data to be included in the attribute; the syntax can be any of the standard syntax types.
tag=*nnn*	Tunnel attributes include a tag field, which may be used to group attributes in the same packet which refer to the same tunnel. Since some vendors' equipment does not support tags, this syntax type is optional and must be present for the attribute to include a tag field. A value of 0 indicates that the field should be present but ignored.

An example of a vendor-specific attribute (VSA) definition is:

```
ATTRIBUTE vsa-xxx 26 [vid=1234 type1=1 len1=+2 data=string] R
```

Flag Characters

The flags setting consists of the concatenation of one or more flag characters from the list in the following table.

Flag Character	Meaning
b *or* B	Indicates that an attribute may be bundled in a single Vendor-Specific-Attribute for a particular vendor id. It may be included as one of a series of subattributes within a single VSA.
c	Attribute can appear once within a user or profile checklist.
C	Attribute can appear multiple times within a user or profile checklist.
r	Attribute can appear once within a user or profile return list.
R	Attribute can appear multiple times within a user or profile return list.
t	Attribute can appear once within a tunnel attribute list.
T	Attribute can appear multiple times within a tunnel attribute list.
o or O	Attribute is orderable; the administrator can control the order in which such attributes are stored in the RSA RADIUS database (this flag makes sense only for multivalued attributes).

VALUE Records

Value records are used to define names for specific integer values of previously defined integer attributes. Value records are never required, but are appropriate where specific meaning can be attached to an integer value of an attribute. The value record must conform to the following syntax:

```
VALUE attrib_name value_name integer_value
```

Parameter	Function
attrib_name	Name of the attribute (up to 31 characters with no embedded blanks)
value_name	Name of the attribute value (up to 31 characters with no embedded blanks)
integer_value	Integer value associated with the attribute value

No two value records in a dictionary file should have the same attrib_name and value_name or the same attrib_name and integer_value. If a duplicate is encountered, the later definition of the attribute value is ignored in favor of the earlier one (the earlier one is considered to be an override).

The following example illustrates the use of the VALUE record to define more user-friendly attribute values for the Framed-Protocol attribute:

```
ATTRIBUTE  Framed-Protocol    7     integer     Cr
VALUE      Framed-Protocol    PPP           1
VALUE      Framed-Protocol    SLIP          2
```

Using these dictionary records, the administrator need not remember that the integer value 1 means PPP and the integer value 2 means SLIP when used in conjunction with the Framed-Protocol attribute. Instead, the Security Console lets you choose from a list of attribute values.

Macro Records

Macro records are used to streamline the creation of multiple vendor-specific attributes that include many common parameters. A macro record can be used to encapsulate the common parts of the record. The macro record must conform to the following syntax:

```
MACRO macro_name(macro_vars) subst_string
```

Parameter	Function
macro_name	Name of the macro
macro_vars	One or more comma-delimited macro variable names
subst_string	String into which macro variables are to be substituted; any sequence of characters conforming to the format %x% for which a macro variable called *x* has been defined undergo the substitution process

The following example illustrates the use of a macro that simplifies the specification of multiple vendor-specific attributes:

```
MACRO Cisco-VSA(t, s) 26 [vid=9 type1=%t% len1=+2 data=%s%]
ATTRIBUTE   Cisco-xxx   Cisco-VSA(1, string)    R
ATTRIBUTE   Cisco-yyy   Cisco-VSA(4, int4)      C
ATTRIBUTE   Cisco-zzz   Cisco-VSA(9, ipaddr)    r
```

The macro preprocessor built into the RSA RADIUS dictionary processing would translate the records in the preceding example to the following records before being processed.

```
ATTRIBUTE Cisco-xxx 26 [vid=9 type1=1 len1=+2 data=string] R
ATTRIBUTE Cisco-yyy 26 [vid=9 type1=4 len1=+2 data=int4]   C
ATTRIBUTE Cisco-zzz 26 [vid=9 type1=9 len1=+2 data=ipaddr] r
```

OPTION Records

By default, each vendor-specific attribute is encoded in a single VSA attribute. The format of a VSA attribute is described in the following table.

Bits	Content
0 - 7	Type: contains the value 26
8 - 16	Length of data in bytes
17 - 47	Vendor ID
48 - on	Vendor data

If you provide a parameter to the OPTION setting, however, multiple vendor-specific attributes can be present in the vendor-data portion of a single VSA record.

The OPTION record must conform to the following format:

```
OPTION bundle-vendor-id = vid
```

Note: You must set the B flag for attribute bundling to occur. For a particular vendor-specific attribute to be bundled, you must set the OPTION record for the vendor’s vendor-ID and set the B (or b) flag for the specific attribute.

The Nortel Rapport dictionary supports this option, for example. If you want to combine Nortel’s vendor-specific attributes in a single VSA, you would provide the entry:

```
OPTION bundle-vendor-id=562
```

This is because 562 is Nortel’s Vendor ID, as set in the MACRO record. The Nortel Rapport vendor-specific attributes now would be concatenated within the vendor-data portion of a RADIUS VSA attribute (up to 249 octets).

classmap.ini File

The **classmap.ini** initialization file specifies what RSA RADIUS does with RADIUS attributes encoded in one or more Class attributes included in accounting requests it receives.

[AttributeName] Section

The [AttributeName] section of **classmap.ini** specifies whether RADIUS information encapsulated in a Class attribute should be appended to an accounting request or replace a current value in an accounting request. If one attribute is replaced by another, the original attribute can be added to the request with a different identifier.

```
[AttributeName]
<add | replace> = Attribute [,Attribute]
```

The following table lists the [AttributeName] parameters and their functions.

Parameter	Function
AttributeName	Name of the attribute encoded into the Class attribute by the authenticating server.
<add | replace>	Specifies whether the attribute value should be added to the accounting request (leaving all other values intact) or whether one value should replace another in the accounting request.
Attribute	Specifies the name of the attribute that should be added to the accounting request, which contains the original value of the attribute identified by *AttributeName*.
[,*Attribute*]	Specifies the name of the attribute in the accounting request that should contain the value of the attribute displaced when the value of *AttributeName* replaces the existing *Attribute* value. Valid only when the replace keyword is used.

Note: The RADIUS Class attribute cannot contain IPv6 attributes.

In the following example, the encapsulated User-Name attribute would replace the existing User-Name in the accounting request.

```
[User-name]
replace = User-Name
```

In the following example, the encapsulated User-Name attribute would be placed in the accounting request as User-Name, and the original value for User-Name would be added to the request as Funk-Full-User-Name.

```
[User-name]
replace = User-Name, Funk-Full-User-Name
```

In the following example, the encapsulated User-Name attribute would be added to the accounting request as a new attribute, and the original User-Name attribute would remain unchanged.

```
[User-Name]
add = Funk-Full-User-Name
```

filter.ini File

Note: Use the Operations Console to maintain settings in the **filter.ini** file. Do not edit the **filter.ini** file manually.

The **filter.ini** configuration file lets you set up rules for filtering attributes into and out of RADIUS packets.

Filter Rules

Each filter in the **filter.ini** file consists of the filter name in square brackets ([*name*]) followed by the rules for that filter.

Each rule takes one of the following three forms:

```
keyword attribute value
keyword attribute
keyword
```

The following table lists valid syntax combinations.

filter.ini Rule Syntax	Function
ALLOW	This keyword by itself specifies that all attributes, regardless of value, are to be allowed in the packet.
ALLOW *attribute*	This rule specifies that this attribute is allowed in the packet, regardless of its value.
ALLOW *attribute value*	The rule lists a specific attribute/value pair to allow in the packet.
EXCLUDE	The keyword by itself specifies that all attributes, regardless of value, are to be excluded from the packet. EXCLUDE is the default action for a filter.
EXCLUDE *attribute*	The rule specifies that this attribute is excluded from the packet, regardless of its value.
EXCLUDE *attribute value*	The rule specifies an attribute/value pair to exclude from the packet.

filter.ini Rule Syntax	Function
ADD *attribute value*	The rule lists a specific attribute/value pair to add to the packet. The attribute is added after all other rules are processed.
REPLACE *attr1* WITH *attr2*	The rule specifies that any occurrence of *attr1* are replaced by *attr2*, which retains *attr1*'s value.
REPLACE *attr1* WITH *attr2 v2*	The rule specifies that any occurrence of *attr1* (regardless of value) is replaced by *attr2* whose value is set to *v2*.
REPLACE *attr1 v1* WITH *attr2*	The rule specifies that any occurrence of *attr1* whose value is *v1* is replaced by *attr2* (which keeps value *v1*).
REPLACE *attr1 v1* WITH *attr2 v2*	The rule specifies that any occurrence of *attr1* whose value is *v1* is replaced by *attr2* having a value *v2*.

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.

The RSA 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, RSA 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 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 **filter.ini** 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. The following table lists the meaning of each attribute type.

Attribute Type	Function
hexadecimal	A hexadecimal value is specified as a string. Special characters may be included using escape codes.
int1, int4, integer	1- or 4-byte unsigned decimal number (integer is equivalent to int4). **Note:** The RSA 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.
ipaddr, ipaddr-pool	An IP address in dotted notation; for example: `EXCLUDE NAS-IP-Address 127.0.0.1`
ipxaddr-pool	A sequence of hex digits; for example: `ALLOW Framed-IPX-Network 0042A36B`

Attribute Type	Function

string	String attribute (includes null terminator). A string is specified as text. The text may be enclosed in double-quotes ("). The text is interpreted as a regular expression. Backslash (\) is the escape character. Escape codes are interpreted as follows:
	Code **Meaning**
	\a 7
	\b 8
	\f 12
	\n 10
	\r 13
	\t 9
	\v 11
	nnn *nnn* is a decimal value between 0 and 255
	\x*nn* *nn* is a hexadecimal value between 00 and FF
	c *c* is a single character, interpreted literally
	Literal backslashes (\) within a string and double-quotes (“) within quoted strings must be prefixed with an escape character. For example:
	`ADD Reply-Message Session limit is one hour`
	`ADD Reply-Message "Session limit is one hour"`
	`ADD Reply-Message "Your username is \"George\""`

time	A time value is specified with a string indicating date and time:
	`yyyy/mm/dd hh:mm:ss`
	The date portion is mandatory; the time portion may be specified to whatever degree of precision is required, or may be omitted entirely. For example:
	`2006/4/3 14:00:00`
	and
	`2006/4/3 14`
	both refer to April 3, 2006 at 2:00 p.m.
	For example:
	`ADD Ascend-PW-Expiration 2006/4/3`

Referencing Attribute Filters

RSA RADIUS attribute filtering provides flexibility in packet processing. 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, *.dir, **peapauth.aut**, or **ttlsauth.aut** file.) Filtering is often used only for packets that are routed “out” to realms (the FilterOut parameter).

To reference the filtering rules defined in the **filter.ini** file in proxy or directed realm configurations, you must use the FilterOut and FilterIn parameters in the [Auth] and [Acct] sections of a RADIUS realm configuration file.

The full syntax used is:

```
[Auth]
FilterIn=name1
FilterOut=name2

[Acct]
FilterIn=name3
FilterOut=name4
```

where *name1*, *name2*, and so forth provide the names of filters, sections in the **filter.ini** file called [*name1*], [*name2*], and so forth. The *name* values in this syntax are completely independent of each other. They may be all the same, all different, or some combination of same and different.

When using the FilterIn and FilterOut parameters in the [Auth] and [Acct] sections, be sure to use the filter name without the square brackets (“name”, not “[name]”).

Note: If a [name] section is not found in the **filter.ini** file, it is equivalent to assigning a filter that EXCLUDEs all attributes. In other words, assigning a filter name that cannot be found causes the final packet to be emptied of all attributes.

Note: Do not allocate IP addresses from RSA RADIUS IP address pools in accounting filters. These addresses will be allocated but never released.

spi.ini File

The **spi.ini** initialization file defines encryption keys and identifies the servers from which RSA RADIUS processes encrypted Class attributes in accounting requests. The **spi.ini** file allows one RSA RADIUS server to decode accounting requests for sessions that were authenticated on a different RSA RADIUS server. Class attributes received from servers not specified in **spi.ini** are ignored.

All RSA RADIUS servers that may receive authentication and accounting requests from a common network access device must be configured with similar **spi.ini** files, which must list the IP addresses of all the servers in that “cluster.” This allows one server to authenticate a user and generate an encrypted Class attribute that can be decrypted and processed by any other server in the cluster.

[Keys] Section

The [Keys] section of **spi.ini** specifies the list of encryption keys used to encode subattributes encapsulated within Class attributes.

```
[Keys]
CurrentKey = n
1 = value
2 = value
M
```

The following table lists the [Keys] parameters and their functions.

Parameter	Function
CurrentKey	Specifies the encryption key that is currently active, where *n* is 0 or the number of a key listed in the [Keys] section: 0 – Generate and use a unique random key to encrypt Class attributes. Used only when the RSA RADIUS server does not exchange encrypted Class attributes with other servers. *n* – Use the specified key to encrypt Class attributes. Default value is 0.
n = *value*	Specifies the number and value of the encryption key.

In the following example, the RSA RADIUS server generates a unique random key to encrypt Class attributes.

```
[Keys]
CurrentKey = 0
```

In the following example, the second key (swordfish) is currently active and used to encrypt Class attributes. The other keys in this section can be used to decrypt Class attributes received from other servers in the same cluster.

```
[Keys]
CurrentKey = 2
1 = firstkey
2 = swordfish
3 = mypassword
```

[Hosts] Section

The [Hosts] section of **spi.ini** identifies the IP address of servers from which received Class attributes are parsed for encapsulated/encrypted subattributes. Class attributes from servers not identified in the [Hosts] section of **spi.ini** are passed without special processing.

The information in the [Hosts] section is used to compute the server's identifier, which is included in the Class attribute. If one of a host's interfaces is included in the [Hosts] section, that interface is used to compute the server identifier. If more than one interface for a host is listed, the IP address of the last interface listed is used. If no matching address is found, the host's primary IP address is used. Addresses not corresponding to a host interface are used to configure the collection of other servers whose Class attributes are accepted.

In the following example, three servers are identified as belonging to a cluster.

```
[Hosts]
192.168.15.21
192.168.23.121
192.168.23.205
```

vendor.ini File

The **vendor.ini** initialization file contains information that allows RSA RADIUS to work with the products of other vendors.

[Vendor-Product Identification] Section

The [Vendor-Product Identification] section of **vendor.ini** identifies and provides information about the network access devices that can be used with RSA RADIUS.

The following table lists the [Vendor-Product Identification] parameters and their functions.

Parameter	Function
vendor-product	Specifies the name of the product. A product name must be unique, cannot include blanks and must consist of 31 or fewer characters. These product names are used only in the Make/model list in the RADIUS Clients panel. This list is used when adding a new RADIUS client or when selecting a vendor-specific attribute.
dictionary	Specifies the dictionary file to use for this product. The dictionary file must be located in the same directory as the RSA RADIUS daemon or service. You do not need to specify an extension on the dictionary name; RSA RADIUS automatically attaches an extension of .DCT to the dictionary names listed in this parameter.
call-filter-attribute	Specifies the attribute used for call filter functions. Used only by Ascend/Lucent network access devices.
challenge-response-attribute	Specifies the attribute number in which a network access device sends responses to challenge sequences. If not specified, the default behavior is to expect responses to be encoded in the User-Password attribute.

Parameter	Function
data-filter-attribute	Specifies the attribute used for data filter functionality. Used only by Ascend/Lucent network access devices.
discard-after	Used for inbound proxy RADIUS servers that send user name information in a “decorated” format. For example, if a proxy RADIUS server sends user names of the form *username@company,* then specifying @ results in the @ delimiter character and all text after the @ delimiter character being discarded for authentication purposes; the string *username* is used.
discard-before	Used for inbound proxy RADIUS servers that send user name information in a “decorated” format. For example, if a proxy RADIUS server sends user names of the form *company$username*, then specifying $ results in the $ delimiter character and all text before the $ delimiter character being discarded for authentication purposes; the string *username* is used.
help-id	Help context for the vendor’s product in the vendor information help file.
ignore-acct-ss	If set to Yes, the digital signature of accounting packets based on the shared secret is ignored. This accommodates devices that do not properly sign accounting packages. Default value is No.
ignore-ports	Determines whether RSA RADIUS may infer that one user has logged off if the port that was assigned to that user is now being used by another user. • If set to No, an inference is made and the previous user is removed from the Active Users list. • If set to Yes, no inference is made and both users are deemed active. Default value is No.
max-eap-fragment	Specifies a maximum EAP fragment length on a make/model basis. The maximum EAP fragment length emitted by TLS or TTLS is the lesser of the maximum specified in their **.eap/.aut** files and this setting. Default value is 1020. This may be inefficient, however, as the fragment length must be set to a number low enough to work with all of a customer's Access Points.

Parameter	Function
port-number-usage	• If set to per-port-type, entries in the Active List containing duplicate port numbers and port types are deleted. • If set to unique, entries in the Active List containing duplicate port numbers are deleted; port type information is ignored. Default value is per-port-type.
product-scan-acct	Specifies the name of the section in the **vendor.ini** file that contains rules for dynamically determining the product associated with an accounting request by the contents of the request packet.
product-scan-auth	Specifies the name of the section in the **vendor.ini** file that contains rules for dynamically determining the product associated with an authentication request by the contents of the request packet.
send-class-attribute	If set to No, the Class attribute is not sent to the client on Access-Accept. (This feature is designed to accommodate devices that don't handle the Class attribute properly.) Default value is Yes.
send-session-timeout-on-challenge	• If set to Yes, the Session-Timeout attribute is sent to the client on Access-Challenge responses that include EAP messages. This attribute advises a network access device on how long it should wait for a user response to the challenge. • If set to No, the Session-Timeout attribute is not sent to the client on Access-Challenge responses that include EAP messages. Default value is Yes.

6 Accounting Configuration Files

This chapter describes the usage and settings for the RSA RADIUS accounting initialization (.ini) file, which enables, disables, and configures accounting features of the server. The account.ini file is loaded at startup time, and resides in the RSA RADIUS directory.

Note: Some parameters or sections may appear in the initialization file that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

Note: If you edit settings in the **accounting.ini** file, you must adhere to the .ini syntax. After making any changes to the **accounting.ini** file, you must restart the RADIUS server for the changes to take effect.

account.ini File

The **account.ini** file contains information that controls how RADIUS accounting attributes are logged to a comma-delimited text file by RSA RADIUS. Specifically, the **account.ini** file controls file creation settings, such as file creation frequency, maximum size, and default directory, and file content, such as what information is recorded for each received accounting request.

[Alias/name] Sections

The [Alias/*name*] sections of **account.ini** are used to associate attributes of different names, but identical meaning. For example, one network access device vendor might call an attribute Acct-Octet-Pkt and another might call it Acct-Oct-Packets, yet the two attributes mean the same thing.

Each [Alias/*name*] section permits you to map one RADIUS accounting attribute that is already being logged by RSA RADIUS to any number of other attributes. You can provide as many [Alias/*name*] sections as you want, using the following syntax for each section:

```
[Alias/name]
VendorSpecificAttribute=
VendorSpecificAttribute=
M
```

The following table lists the [Alias/name] parameters and their functions.

Parameter	Function
name	The preferred attribute name. The name attribute must be one that you are currently logging to a column in the RSA RADIUS accounting log file (.act). Therefore, it must be listed in the [Attributes] section of **account.ini**.
VendorSpecificAttribute	Each entry is given on one line. An equal sign (=) must immediately follow each VSA name, without any intervening space. Improperly formatted entries are considered invalid and are ignored.

Each *VendorSpecificAttribute* in the list is logged to the *name* column in the accounting log file. Because you are listing these attributes in an [Alias/*name*] section, verify they are not listed in the [Attributes] section, or they will be logged to their own columns as well as the *name* column.

All of the attribute names that you reference in an [Alias/*name*] section must be defined in a dictionary file that is already installed on the RSA RADIUS server. This includes *name* and each *VendorSpecificAttribute* entry.

In the following example, the standard RADIUS attribute Acct-Octet-Packets is mapped to the vendor-specific attributes Acct-Octet-Pkt and Acct-Oct-Packets. Values encountered for all three attributes are logged in the Acct-Octet-Packets column in the accounting log file:

```
[Alias/Acct-Octet-Packets]
Acct-Octet-Pkt=
Acct-Oct-Packets=
```

[Attributes] Section

The [Attributes] section of the **account.ini** file lists all the attributes logged for each received accounting request in the accounting log file. When you install RSA RADIUS, the **account.ini** file is set up so that all standard RADIUS attributes and all supported vendors' accounting attributes are listed.

You can change the order of columns in the accounting log file by rearranging the sequence of attributes in the [Attributes] section. You can delete or comment out any attributes that are not relevant to your billing system or which do not apply to the equipment that you are using. This lets you design the content and column order of any spreadsheets that you plan to create based upon the accounting log file.

The syntax is as follows:

```
[Attributes]
AttributeName=
AttributeName=
M
```

For example:

```
[Attributes]
User-Name=
NAS-Port=
Framed-IP-Address=
Acct-Status-Type=
Acct-Delay-Time=
Acct-Session-Id=
```

The [Attributes] section lists one *AttributeName* on each line. You must ensure that an equal sign (=) immediately follows each *AttributeName*, with no spaces in between. Improperly formatted entries are considered invalid and are ignored.

Each *AttributeName* in the [Attributes] section must be defined in a standard RADIUS dictionary file or a vendor-specific dictionary file on the RSA RADIUS server.

Note: The first six attributes in each log file entry (Date, Time, RAS-Client, Record-Type, Full-Name, and Auth-Type) are always enabled, and cannot be re-ordered or deleted. Therefore, these attributes do not appear in the **account.ini** file [Attributes] section.

[Settings] Section

RSA RADIUS writes all accounting data to the current accounting log file (.act) until that log file is closed. After closing the file, RSA RADIUS opens a new one and begins writing accounting data to it. You can configure how often this rollover of the accounting log file occurs.

The naming conventions for accounting log files permit more than one file to be generated during a day. The following table lists the file naming conventions used for different rollover periods. In the examples below, *y*=year digit, *M*=month digit, *d*=day digit, *h*=hour digit, and *m*=minute digit. When more than one file is generated during a day, the sequence number _*nnnnn* starts at _00000 each day.

File Generation Method	File Naming Convention
Default (24 hours)	*yyyyMMdd*.act
Non-24-hour rollover	*yyyyMMdd_hhmm*.act
Rollover due to size	*yyyyMMdd_nnnnn*.act
Rollover due to size or startup when non-24-hour time in effect	*yyyyMMdd_hhmm_nnnnn*.act

The [Settings] section of the **account.ini** file controls how entries are written to the accounting log file, and ensures the compatibility of these entries with a variety of database systems.

The following table lists the [Settings] parameters and their functions.

Parameter	Function
BufferSize	The size of the buffer used in the accounting logging process, in bytes. Default value is 131072 bytes.
Carryover	• If set to 1, each time a new accounting log file is created, a start record for each session that is currently active is written to the file. • If set to 0, the list is not written. Default value is 1.
Enable	• If set to 1, the accounting log feature is enabled. • If set to 0, no .act files are created on this server. Accounting servers must have Enable set to 1; for efficiency, non-accounting servers must have Enable set to 0. Default value is 1.
LineSize	Number in the range 1024–32768 that specifies the maximum size of a single accounting log line. Default value is 4096.
LogFilePermissions (Solaris/Linux only)	Specifies the owner and access permission setting for the accounting log file. Enter a value for the LogFilePermissions setting in *owner:group permissions* format, where: • *owner* specifies the owner of the file in text or numeric format. • *group* specifies the group setting for the file in text or numeric format. • *permissions* specifies what privileges can be exercised by Owner/Group/Other with respect to the file in text or numeric format. For example, ralphw:1007 rw-r----- specifies that the file owner (ralphw) can read and edit the log file, members of group 1007 can read (but not edit) the log file, and that other users cannot access the log file.
MaxSize	The maximum size of an accounting log file, in bytes. If the accounting log file reaches or exceeds this size when it is checked, the log file is closed and a new file started. A value of 0 (the default) means unlimited size. **Note:** Because the size of the log file is checked once per minute, the log file can exceed the maximum size specified in this parameter.

Parameter	Function
QuoteBinary	• If set to 1, binary values written to the accounting log file are enclosed in quotes. • If set to 0, quotes are not used. Set this value according to the format expected by the accounting application that processes the entries. Default value is 1.
QuoteInteger	• If set to 1, integer values written to the accounting log file are enclosed in quotes. • If set to 0, quotes are not used. Set this value according to the format expected by the accounting application that processes the entries. Default value is 1.
QuoteIPAddress	• If set to 1, IP addresses written to the accounting log file are enclosed in quotes. • If set to 0, quotes are not used. Set this value according to the format expected by the accounting application that processes the entries. Default value is 1.
QuoteText	• If set to 1, text strings written to the accounting log file are enclosed in quotes. • If set to 0, quotes are not used. Set this value according to the format expected by the accounting application that processes the entries. Default value is 1.
QuoteTime	• If set to 1, time and date values written to the accounting log file are enclosed in quotes. • If set to 0, quotes are not used. Set this value according to the format expected by the accounting application that processes the entries. Default value is 1.
RollOver	Specifies how often the current accounting log file is closed and a new file opened (a rollover), up to one rollover per minute. Non-zero values indicate the number of minutes until the next rollover. If set to 0, the accounting log file rolls over once every 24 hours, at midnight local time. Default value is 0.

Parameter	Function
RollOverOnStartup	• If set to 1, each time RSA RADIUS is started, it closes the current accounting log file and opens a new one. A sequence number _*nnnnn* is appended to the log file name, just as when MaxSize is reached. • If set to 0, each time RSA RADIUS is started, it appends entries to the previously open accounting log file. Default value is 0.
Titles	• If set to 1, each time a new accounting log file is created, the title line (containing column headings) is written to the file. • If set to 0, the line is not written. Default value is 1.
UTC	• If set to 1, time and date values are provided according to Universal Time Coordinates (UTC, formerly known as Greenwich Mean Time or GMT). • If set to 0, time and date values reflect local time. Default value is 0.

[TypeNames] Section

Each entry in the [TypeNames] section of **account.ini** maps a possible value of the Acct-Status-Type attribute to a string. The value of this attribute is written into the fourth column of each accounting log record.

The syntax is as follows:

```
[TypeNames]
TypeID = TypeName
TypeID = TypeName
M
```

The following table lists the [TypeNames] parameters and their functions.

Parameter	Function
TypeID	Each *TypeID* is a numeric value that corresponds to a possible value of the Acct-Status-Type attribute. This attribute appears in every incoming RADIUS accounting packet to identify the types of data it is likely to contain.
TypeName	Each *TypeName* value is a string. This string is written to the accounting log to identify the type of packet.

The standard Acct-Status-Type values 1, 2, 3, 7, and 8 are already listed in the [TypeNames] section of **account.ini** as follows:

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
```

You can edit the [TypeNames] section to add vendor-specific packet types to this list, which makes your accounting log files easier to read and use. For example:

```
[TypeNames]
1=Start
2=Stop
3=Interim
7=On
8=Off
639=AscendType
28=3ComType
```

If no string is given for a particular Acct-Status-Type, RSA RADIUS uses the numeric value of the incoming Acct-Status-Type attribute, formatted as a string.

7 EAP Configuration Files

This chapter describes the EAP configuration and helper files, which specify options for automatic EAP helper methods. These files are loaded at startup time and reside in the RSA RADIUS directory.

Note: Some parameters or sections may appear in the configuration and helper files that are not used for this release of RSA RADIUS. The descriptions in this document address those sections or parameters that are relevant for this release. Do not modify parameters that are not described in this document.

eap.ini File

Note: Use the RSA Operations Console to maintain settings in the **eap.ini** file. Do not edit the **eap.ini** file manually.

The **eap.ini** configuration file configures the sequence in which EAP authentication types are tried when authenticating users by means of the different RSA RADIUS authentication methods.

Each authentication method that you want EAP authentication to be performed against must be configured within this **eap.ini** file.

This file must contain one section for each authentication method that you use, and the title of the section must identify the authentication method:

- SecurID
- EAP-TTLS
- EAP-PEAP

The following table lists the parameters in each section.

Parameter	Function
EAP-Only	• If set to 0, the authentication method accepts all types of user credentials. • If set to 1, the authentication method is given only EAP credentials or acts only as a back-end server to an automatic EAP protocol method. For authentication methods expected to handle EAP-TTLS inner authentications, this parameter must be set to 0 or 1 depending on the type of credentials used in the inner authentication. **Note:** If you are using SecurID with PEAP, set this value to 0. Since the PEAP plug-in converts the inner EAP/Generic Token credentials to PAP for security reasons, setting this value to 1 causes SecurID processing to be skipped when using EAP/Generic Token, ultimately leading to the user being rejected.
EAP-Type	A comma-separated list of the EAP protocols to support this authentication method. The first protocol in the list is the primary protocol. Protocols that appear later in the list are used with this authentication method only if the client responds with an EAP NAK and specifies such a protocol or if another authentication method triggers the use of the protocol but cannot complete the request. Valid values include the following: • SecurID – EAP-15 – EAP-32 – Generic-Token • EAP-TTLS – TTLS • EAP-PEAP – PEAP Leave the EAP-Type list empty to disable EAP for this authentication method.

Parameter	Function
First-Handle-Via-Auto-EAP	• If set to 1 and the user credentials are EAP, an appropriate automatic EAP helper method is called before the authentication method. The purpose of calling the automatic EAP helper method is to convert the user's EAP credentials into a format acceptable to the authentication method. • If set to 0, the authentication method itself handles the request directly, before any automatic helper methods. Default varies based on type of user. Refer to the comments in the **eap.ini** file for more information. **Note:** If you want to use machine authentication, you must enter **1** for this setting in the [Windows Domain User] and [Windows Domain Group] sections of **eap.ini**. **Note:** You must set the AllowMachineLogin setting in the [WindowsDomain] section of **winauth.aut** to **Yes** if you want to use machine authentication.
Available-EAP-Types	A comma-separated list of the EAP protocols that can be selected when configuring the RSA RADIUS server by means of the Security Console. Valid values include the following: • TTLS • Generic-Token

peapauth.aut File

Note: If you edit settings in the **peapauth.aut** file, you must adhere to standard .ini syntax. After making any changes to the **peapauth.aut** file, you must restart the RADIUS server for the changes to take effect.

Settings for the EAP-PEAP plug-in are stored in the **peapauth.aut** file. The **peapauth.aut** configuration file is read each time the RSA RADIUS server restarts.

[Bootstrap] Section

The [Bootstrap] section of the **peatauth.aut** file specifies information that RSA RADIUS uses to load the EAP-PEAP authentication method.

The following table lists the [Bootstrap] parameters and their functions.

Parameter	Function
LibraryName	Specifies the name of the EAP-PEAP module. Default value is **peapauth.dll** for Windows and **peapauth.so** for Solaris and Linux. Do not change this unless you are advised to do so by RSA Customer Support.
Enable	Specifies whether the EAP-PEAP authentication module is enabled. • If set to 0, EAP-PEAP is disabled, and the authentication method does not appear in the Authentication Methods list in the Authentication Policies panel. • If set to 1, EAP-PEAP is enabled. Default value is 0.
InitializationString	Specifies the name of the authentication method to appear in the Authentication Methods list in the Authentication Policies panel. The name of each authentication method must be unique. If you create additional .aut files to implement authentication against multiple databases, the InitializationString value in each file must specify a unique method name. Default value is EAP-PEAP.

[Server_Settings] Section

The [Server_Settings] section lets you configure the basic operation of the EAP-PEAP plug-in.

The following table lists the [Server_Settings] parameters and their functions.

Parameter	Function
TLS_Message_Fragment_Length	Set to the maximum size TLS message length that may be generated during each iteration of the TLS exchange. 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 value (1020) prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame. This is likely to be the safest setting. Setting a smaller value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. While a value of 1400 may result in 6 round-trips, a value of 500 may result in 15 round-trips. The minimum value is 500.

Parameter	Function
Return_MPPE_Keys	Setting this attribute to 1 causes the module to include RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Accept response sent to the Access Point. This is necessary for the Access Point to key the WEP encryption. If the Access Point is authenticating only end users and WEP is not being used, this attribute may be set to 0. Default value is 1.
DH_Prime_Bits	Specifies the size of the prime number that the module uses for Diffie-Hellman exponentiation. Selecting a larger 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. Default value is 1024.
Cipher_Suites	Specifies the TLS cipher suites (in order of preference) that the server is to use. 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, and 0x09.
PEAP_Min_Version	Specifies the minimum version of the PEAP protocol that the server should negotiate: • If set to 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1). • If set to 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU). Default value is 0. **Note:** The value entered in this setting must be less than or equal to the value entered for the PEAP_Max_Version setting.

Parameter	Function
PEAP_Max_Version	Specifies the maximum version of the PEAP protocol that the server should negotiate: • If set to 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1). • If set to 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU). Default value is 1. **Note:** The value entered in this parameter must be equal to or greater than the value entered for PEAP_Min_Version.

[Session_Resumption] Section

The [Session_Resumption] section lets you specify whether session resumption is permitted and under what conditions 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.

The following table lists the [Session_Resumption] parameters and their functions.

Parameter	Function
Session_Timeout	Set this attribute to the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate. • If set to a number greater than 0, the lesser of this value and the remaining resumption limit (see description below) is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access Accept response. • If set to 0, no Session-Limit attribute is generated by the plug-in. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute. Default value is 0. 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.

Parameter	Function
Termination_Action	Set this attribute to the integer value that you want returned in a Termination-Action attribute. This is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. If you do not specify a value for this attribute, the 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. Default is to not send this attribute.
Resumption_Limit	Set this attribute to the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature. 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. Default value is 0.

ttlsauth.aut File

Note: If you edit settings in the **ttlsauth.aut** file, you must adhere to standard .ini syntax. After making any changes to the **ttlsauth.aut** file, you must restart the RADIUS server for the changes to take effect.

Settings for the EAP-TTLS authentication method are stored in the **ttlsauth.aut** file. The **ttlsauth.aut** configuration file is read each time the RSA RADIUS server restarts.

[Bootstrap] Section

The [Bootstrap] section of the **ttlsauth.aut** file specifies information that RSA RADIUS uses to load the EAP-TTLS authentication method.

The following table lists the [Bootstrap] parameters and their functions.

Parameter	Function
LibraryName	Specifies the name of the EAP-TTLS module. Default value is ttlsauth.dll for Windows and ttlsauth.so for Solaris and Linux. Do not change this unless you are advised to do so by Juniper Customer Support.

Parameter	Function
Enable	Specifies whether the EAP-TTLS authentication module is enabled. • If set to 0, EAP-TTLS is disabled, and the EAP-TTLS authentication method does not appear in the Authentication Methods list in the Authentication Policies panel. • If set to 1, EAP-TTLS is enabled. Default value is 0.
InitializationString	Specifies the name of the authentication method to appear in the Authentication Methods list in the Authentication Policies panel. The name of each authentication method must be unique. If you create additional .aut files to implement authentication against multiple databases, the InitializationString value in each file must specify a unique method name. Default value is EAP-TTLS.

[Server_Settings] Section

The [Server_Settings] section lets you configure the basic operation of the EAP-TTLS plug-in.

The following table lists the [Server_Settings] parameters and their functions.

Parameter	Function
TLS_Message_Fragment_Length	Specifies the maximum size TTLS message length that may be generated during each iteration of the TTLS exchange. 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). Minimum value is 500. Maximum value is 4096. Default value is 1020, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

Parameter	Function
Return_MPPE_Keys	Setting this attribute to 1 causes the module to include RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Accept response sent to the Access Point. This is necessary for the Access Point to key the WEP encryption. If the Access Point is authenticating only end users and WEP is not being used, this attribute may be set to 0. Default value is 1.
DH_Prime_Bits	Specifies the size of the prime number that the module uses for Diffie-Hellman exponentiation. Selecting a larger 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. Default value is 1024.
Cipher_Suites	Specifies the TLS cipher suites (in order of preference) that the server is to use. 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, and 0x09.
Require_Client_Certificate	• If set to 1, specifies that the client must provide a certificate as part of the TTLS exchange. • If set to 0, no client certificate is required. Default value is 0.

[Session_Resumption] Section

The [Session_Resumption] section lets you specify whether session resumption is permitted and under what conditions 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 re-authenticate after the session timer has expired.

The following table lists the [Session_Resumption] parameters and their functions.

Parameter	Function
Session_Timeout	Set this attribute to the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate. • If set to a number greater than 0, the lesser of this value and the remaining resumption limit (see description below) is sent in a Session-Limit attribute to the network access device on the RADIUS Access Accept response. • If set to 0, no Session-Limit attribute is generated by the plug-in. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute. Default value is 0. 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.
Termination_Action	Set this attribute to the integer value that you want returned in a Termination-Action attribute. This is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. If you do not specify a value for this attribute, the 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. Default is to not send this attribute.
Resumption_Limit	Set this attribute to the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature. 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. Default value is 0.

[Integrity_Settings] Section

The [Integrity_Settings] section specifies the list of quarantine profiles that can be used by the optional Endpoint Assurance Server software to specify how to process users designated for isolation.

```
[Integrity_Settings]
;Quarantine_Profiles=QUARANTINE QUARANTINE2
```

The following table describes the [Integrity_Settings] parameter and its function.

Parameter	Function
Quarantine_Profiles	Identifies the list of RSA RADIUS profiles that can be assigned to users designated for isolation by the Endpoint Assurance Server software. To enter more than one profile name, enter each name on the same line, separating the profile names with a space. Default value is no quarantine profiles.

Sample ttlsauth.aut File

```
[Bootstrap]
LibraryName=ttlsauth.dll
Enable=1
InitializationString=EAP-TTLS

; Maximum TLS Message fragment length EAP-TLS will handle.
TLS_Message_Fragment_Length = 1020

; Indicates whether the EAP-TLS module should return the
; MS-MPPE-Send-Key and MS-MPPE-Recv-Key attribute upon
successful
; authentication of user.
Return_MPPE_Keys = 1

; Size of the prime to use for DH modular exponentiation.
DH_Prime_Bits = 1536
; TLS cipher suites (in order of preference)
; that the server is to use.
Cipher_Suites = 0x16, 0x13, 0x66, 0x15, 0x12, 0x0a, 0x05,
0x04, 0x07, 0x09
```

Index

A
Accept, 12
account.ini
  Alias/name, 57
  Attributes, 58
  Settings, 59
  TypeNames, 62
Acct section
  in *.pro files, 52
Acct-Status-Type, 63
ADD, 49
AddDestIPAddressAttrToRequest, 9
AddDestUDPPortAttrToRequest, 10
AddSourceIPAddressAttrToRequest, 10
Alias/name section
  in account.ini, 57
ALLOW, 48
AllowSystemPins, 34
Apply-Login-Limits, 10
Attributes section
  in account.ini, 58
Auth section
  in *.pro files, 52
AuthenticateOnly, 10
Available-EAP-Types, 67

B
Bootstrap section
  in peapauth.aut, 67
  in ttlsauth.aut, 71
BufferSize, 60

C
CachePasscodes, 24
call-filter-attribute, 54
Carryover, 60
challenge-response- attribute, 54
CheckMessageAuthenticator, 11
CheckUserAllowed ByClient, 34
Cipher_Suites, 69, 73
class attribute, 16
ClassAttributeStyle, 11
classmap.ini, 18, 47
cluster, for class attribute encryption, 52
Configuration section
  in radius.ini, 9
Current Sessions Table, 17
CurrentKey, 53
CurrentSessions section
  in radius.ini, 17

D
data-filter-attribute, 55
DefaultProfile, 34
DH_Prime_Bits, 69, 73
Dictionary, 54
Diffie-Hellman, 69, 73
DisableSecondaryMakeModelSelection, 12
discard-after, 55
discard-before, 55

E
eap.ini, 65
EAP-15, 34
EAP-32, 34
EAP-Only, 66
EAP-Type, 66
EmbedInClass section
  in radius.ini, 18
Enable, 60, 68, 72
encryption key, 52
EnhancedDiagnosticLogging, 12
EOTP, 34
EXCLUDE, 48
exponentiation, 69, 73
Extended One-Time Password, 34

F
FailedAuthOriginStats section
  in radius.ini, 18
file permissions, 29
filter rules, 48
First-Handle-Via-Auto-EAP, 67
FramedIPAddressHint, 12

G
Greeting, 34

H
hex4, 43
hexadecimal, 43
HiddenEAPIdentity section
  in radius.ini, 19
Hosts section
  in spi.ini, 53

I

Ignore-Acct-SS, 55
ignore-ports, 55
InitializationString, 68, 72
int1, 43
int4, 43
integer, 43
IP address pool, 16
ipaddr, 43
ipaddr-pool, 43
IPv6 section
    in radius.ini, 20
ipxaddr-pool, 43

K

Keys section
    in spi.ini, 53

L

LibraryName, 68, 71
license ID
    determining, 6
LineSize, 60
log file
    size limit, 13
LogAccept, 12
LogFileMaxMBytes, 13
LogFilePermissions, 13, 60
LogHighResolutionTime, 13
LogLevel, 14
LogReject, 14

M

machine authentication, 67
macro records, 45
Max-EAP-Fragment, 55
MaxSize, 60
MS-CHAP
    name stripping, 21
MsChapNameStripping, 21

N

name stripping, MS-CHAP, 21
negative number in attribute, 43
New PIN mode, 34
NoNullTermination, 14
null terminator, 43

P

PEAP_Max_Version, 70
PEAP_Min_Version, 69
peapauth.aut, 51, 67
permissions, 29
PhantomTimeout, 14
PIN, 34
PIN, system-generated, 34
port-number-usage, 56
Ports section
    in radius.ini, 22
POTP, 34
PrivateDir, 14
product-scan-acct, 56
product-scan-auth, 56
Protected One-Time Password, 34
ProxySource, 15
ProxyStripRealm, 15

Q

Quarantine_Profiles, 75
QuoteBinary, 61
QuoteInteger, 61
QuoteIPAddress, 61
QuoteText, 61
QuoteTime, 61

R

RADIUS, 29
radius.ini, 7, 32
    Configuration, 9
    CurrentSessions, 17
    EmbedInClass, 18
    FailAuthOriginStats, 18
    HiddenEAPIdentity, 19
    IPv6, 20
    Ports, 22
    SecurID, 24
RADIUS_HIGH_FDS, 29
RADIUS_PRIVATE_DIR, 29
RADIUSARGS, 29
RADIUSMASK, 29
RADIUSOPTS, 29
Reject, 14
REPLACE, 49
Require_Client_Certificate, 73
Return_MPPE_Keys, 35
Rollover, 61
RolloverOnStartup, 62
RSA RADIUS, 7

S

SecondsToCachePasscodes, 24
SecureTcpAdminPort, 23
SecurID section
  in radius.ini, 24
securid.ini, 33
Send-Class-Attribute, 56
SendOnlyOneClassAttribute, 15
send-session-timeout-on-challenge, 56
server log file
  size limit, 13
Settings section
  in account.ini, 59
signed integer, as attribute type, 43
signed-integer, 43
size limit in server log, 13
spi.ini, 52
StartupTImeout, 16
string, 43
stringnz, 43
system-generated PIN, 34

T

TCPControlAddress, 23
TCPControlPort, 23
time, 43
time attribute, 43
Titles, 62
TraceLevel, 16
transforming user names, 24
TreatAddressPoolsAsDisjoint, 16
ttlsauth.aut, 51, 71
TypeNames section
  in account.ini, 62

U

UDPAcctPort, 23
UDPAuthPort, 23
UDPProxyPortBlockLength, 23
UDPProxyPortBlockStart, 23
ULIMIT_CORE_COUNT, 28
ULIMIT_CORE_SIZE, 28
ULIMIT_OPEN_FILES, 28
umask, 29
unsigned integer, as attribute type, 43
UseNewAttributeMerge, 17
user names
  transforming, 24
UTC, 62

V

vendor.ini, 54
version number, determining, 6

W

WATCHDOG, 30
WATCHDOGARGS, 30
WATCHDOGENABLE, 30
WATCHDOGOPTS, 30

Y

yyyymmdd.log, 13