CHAPTER 3

Configuring the RADIUS Listener LEG

This module describes the configuration procedure for the RADIUS Listener LEG.

The RADIUS Listener LEG is configured using the SM configuration file **p3sm.cfg**, which resides in the **sm-inst-dir/sm/server/root/config** directory (**sm-inst-dir** refers to the SM installation directory).

The configuration file consists of sections headed by a bracketed section title; for example, [Radius.Subscriber ID]. Each section consists of several parameters having the format **parameter=value**. The number sign (“#”) at the beginning of a line signifies that it is a remark.

The General RADIUS Listener LEG configuration settings reside in the [Radius Listener] section. All additional RADIUS Listener LEG sections start with the prefix **Radius.**, such as **[Radius.NAS.nas1]**, and they are defined initially as remark lines.

- Configuring the General Settings, page 3-4
- Information About the Regular Expression Utility, page 3-5
- Configuring RADIUS Attributes Mapping, page 3-8
- Configuring the NAS Devices, page 3-16

Configuring the General Settings

[Radius Listener] Section

The **[Radius Listener]** section in the SM configuration file contains the following parameters:

- start

  Defines whether the SM should run the RADIUS Listener at startup.

  Possible values for this parameter are **yes** and **no**. The default value is **no**.

- accounting_port

  Defines the RADIUS Listener accounting port number.

  The default value is 1813.

- ip

  The IP address to which the RADIUS listener should bind. Use this parameter only in cases where the IP address used for RADIUS transactions is not the main IP address of the SM machine. (For example in an SM cluster.)

  Possible values are any IP address in dotted notation. The default value is not set.

- packet_types

  Defines the RADIUS protocol packet types to analyze.

  Possible values are **accounting-start**, **accounting-interim**, **accounting-stop** separated by a comma.

  The default value is **accounting-start,accounting-interim,accounting-stop**.

Example

The following example is a portion of a configuration file illustrating the **[Radius Listener]** section:

```
[Radius Listener]
# The following parameter defines whether the SM should
# run the RADIUS Listener at startup.
# Receives the values: yes, no. (default no)
start=no
# accounting port number (default 1813)
accounting_port=1813
# RADIUS packet types
packet_types=accounting-start,account-interim,accounting-stop
```

Information About the Regular Expression Utility

A regular expression consists of a character string where some characters are given special meaning with regard to pattern matching. Regular expressions have been in use from the early days of computing, and provide a powerful and efficient way to parse, interpret and search and replace text within an application.

- Supported Syntax, page 3-5
- Unsupported Syntax, page 3-7
- Regular Expression Examples, page 3-7

Supported Syntax

Within a regular expression, the following characters have special meaning:

- Positional Operators
  - **^** matches at the beginning of a line
  - **$** matches at the end of a line
  - **\A** matches the start of the entire string
  - **\Z** matches the end of the entire string
  - **\b** matches at a word break (Perl5 syntax only)
  - **\B** matches at a non-word break (opposite of \b) (Perl5 syntax only)
  - **\<**matches at the start of a word (egrep syntax only)
  - **\>**matches at the end of a word (egrep syntax only)
- One-Character Operators
  - **.** matches any single character
  - **\d** matches any decimal digit
  - **\D** matches any non-digit
  - **\n** matches a newline character
  - **\r** matches a return character
  - **\s** matches any whitespace character
  - **\S** matches any non-whitespace character
  - **\t** matches a horizontal tab character
  - **\w** matches any word (alphanumeric) character
  - **\W** matches any non-word (alphanumeric) character
  - **\x** matches the character x, if x is not one of the above listed escape sequences
- Character Class Operator
  - **[abc]** matches any character in the set **a**, **b**, or **c**
  - **[^abc]** matches any character not in the set **a**, **b**, or **c**
  - **[a-z]** matches any character in the range **a** to **z**, inclusive
  - A leading or trailing dash will be interpreted literally.

- Within a character class expression, the following sequences have special meaning if the syntax bit RE_CHAR_CLASSES is on:
  - **[:alnum:]** Any alphanumeric character
  - **[:alpha:]** Any alphabetical character
  - **[:blank:]** A space or horizontal tab
  - **[:cntrl:]** A control character
  - **[:digit:]** A decimal digit
  - **[:graph:]** A non-space, non-control character
  - **[:lower:]** A lowercase letter
  - **[:print:]** Same as graph, but also space and tab
  - **[:punct:]** A punctuation character
  - **[:space:]** Any whitespace character, including newline and return
  - **[:upper:]** An uppercase letter
  - **[:xdigit:]** A valid hexadecimal digit
- Subexpressions and Backreferences
  - **(abc)** matches whatever the expression **abc** would match, and saves it as a subexpression. Also used for grouping.
  - **(?:...)** pure grouping operator, does not save contents
  - **(?#...)** embedded comment, ignored by engine
  - **\n** where 0 < **n** <10, matches the same thing the nth subexpression matched.
- Branching (Alternation) Operator
  - **a**|**b** matches whatever the expression **a** would match, or whatever the expression **b** would match.
- Repeating Operators

  These symbols operate on the previous atomic expression.
  - **?** matches the preceding expression or the null string
  - ***** matches the null string or any number of repetitions of the preceding expression
  - **+** matches one or more repetitions of the preceding expression
  - **{m}** matches exactly **m** repetitions of the one-character expression
  - **{m,n}** matches between **m** and **n** repetitions of the preceding expression, inclusive
  - **{m,}** matches **m** or more repetitions of the preceding expression
- Stingy (Minimal) Matching

  If a repeating operator is immediately followed by a **?**, the repeating operator will stop at the smallest number of repetitions that can complete the rest of the match.

- Lookahead

  Lookahead refers to the ability to match part of an expression without consuming any of the input text. There are two variations to this:

  – **(?=foo)** matches at any position where **foo** would match, but does not consume any characters of the input.

  – **(?!foo)** matches at any position where **foo** would not match, but does not consume any characters of the input.

Unsupported Syntax

Some flavors of regular expression utilities support additional escape sequences, and this is not meant to be an exhaustive list of unsupported syntax.

- **(?mods)** inlined compilation/execution modifiers (Perl5)
- **\G** end of previous match (Perl5)
- **[.symbol.]** collating symbol in class expression (POSIX)
- **[=class=]** equivalence class in class expression (POSIX)
- **s/foo/bar/** style expressions as in sed and awk (note: these can be accomplished through other means in the API)

Regular Expression Examples

The following examples show how to perform some basic manipulation using the regular expression tool.

- In order to remove the “cisco.com” suffix from a given name (e.g. name@cisco.com), define the following manipulation rule: **(.*)@.***
- In order to remove the “name” prefix from a given name (e.g. name@cisco.com), define the following manipulation rule: **.*@(.*)**
- If no reduction is needed, either define the following reduction rule **(.*)** , or do not define any rule.
- In order to find a partial match from of a name (e.g. find 'isco' in name@cisco.com), define the following matching rule: **mapping_table.cisco=<value>**
- In order to find a full match of a name, define the following matching rule: **mapping_table.cisco=<value>**
- In order to define a default value for an empty or non-existent attribute value: **mapping_table.^$=<value>**

Note Brackets '[' or ']' are used by the regular expression mechanism and also in the section definitions in the configuration files. If either of these characters form a part of the reduction rule definition (**field_manipulation** parameter) or in matching rule definition (**mapping_table** parameter), it should be preceded by two backslashes ('\\') in order to be used as part of the value.

Note For example,

```
[Sample Section]
mapping_table.^12\\[user7$=7 will have a full match result to the '12\[user7' string
mapping_table.^12\[user7$=7 will have a full match result to the '12[user7' string
mapping_table.^12[user7$=7 will have a full match result to the '12[user7' string as well.
```

Note Parentheses '(' or ')' and the pipe character '|' should be preceded by a single backslash '\'. For example, if you have the rule **^(?:88|99)(.*)$ --input=886**, the CLU command should be as shown in the following example:

```
p3radius --test-reduction-rule --reg-exp=^\(?:88\|99\)\(.*\)$ --input=886
```

Note Using complex regular expression patterns will cause performance degradation; the degradation increases relative to the length of the string being manipulated. It is strongly recommended **not** to use the '`[]`' operands with the **.*** or with the **(.*)** operands.

Configuring RADIUS Attributes Mapping

- Mapping of RADIUS Attribute to Subscriber ID, page 3-8
- Mapping of RADIUS Attribute to Subscriber Policy, page 3-11
- Mapping of RADIUS Attribute to Subscriber IP Address, page 3-14

Mapping of RADIUS Attribute to Subscriber ID

Note The configuration described in this section is optional.

The subscriber ID is usually put in the User-Name RADIUS attribute. However, in certain installations, it is possible to use a different RADIUS attribute. For example, in wireless environments, it is possible to use the 3GPP-IMSI or the 3GPP2-IMSI attributes. The default is to use the User-Name attribute.

The RADIUS Listener can be configured to concatenate several RADIUS attributes for use as the Subscriber ID.

To define which attribute(s) to use for the subscriber ID, configure the **[Radius.Subscriber ID]** and the **[Radius.Field.<field name>]** sections. To define the attribute(s) to use, configure the following parameters in the **[Radius.Subscriber ID]** section:

- fields

  Defines the RADIUS protocol fields names. When defining multiple fields, use commas between the field names. The field name must not start or end with a space character and it cannot contain an '=' character. A maximum of three fields can be defined.

  The default value is **user_name**.

  The following is an example of setting this parameter:

  ```
  fields=user_name,vpn
  ```

- field_separator

  Defines the character or string to be used when concatenating several fields.

  In you define three values for the **fields** parameter (user_name, vpn, and IP) and the field separator is being defined:

  - The **field_separator** parameter must contain user_name, vpn, and IP in the same order as they were defined in the **fields** parameter.
  - The separator character between the 1st and 2nd attributes and between the 2nd and 3rd attributes can be different; for example, **user_name-vpn::IP**
  - The separator can be string; for example, **::**
  - The separator can be an empty string; for example, **field_separator=user_namevpnIP**

- The default value is _.

  The following is an example of setting this parameter with the value '-':

  ```
  field_separator=user_name-vpn
  ```

- field_manipulation.<field name>=<regular expression>

  The **field_manipulation** parameters define how to manipulate the RADIUS field values.

  The <field name> part of this parameter is one of the fields defined in the **fields** parameter. The <regular expression> part of this parameter is the reduction regular expression to be used on the <field name> value.

  It is possible to define a field_manipulation rule for each name in the field property. The following is an example of setting this parameter:

  ```
  field_manipulation.user_name=(.*)@.*
  field_manipulation.vpn=(.*)
  ```

  It is possible to configure the RADIUS listener to strip a RADIUS attribute based on a selected character or string using a regular expression rule. This provides a convenient method for obtaining the subscriber ID from a prefix or a suffix of an attribute value.

  For example, you can obtain the subscriber ID from the USERNAME attribute value of subscriber@domain-name by stripping the characters from the value by using the **(.*)@.*** regular expression rule to produce the subscriber. Similarly, you can obtain the domain name by using the **.*@(.*)** regular expression rule.

For each field defined by the **fields** parameter, you must also define a **[Radius.Field.<field name>]** section with the following parameters:

- radius_attribute

  Configure the **radius_attribute** parameter with the RADIUS attribute number. Use the following format for Vendor Specific Attributes (VSA): 26(vendor-id;sub-attribute). For example, **26(10415;1)**.

  The default value is **-1**.

- radius_attribute_type

  Configure **radius_attribute_type** parameter according to the RADIUS attribute format.

  Possible values for this parameter are **integer** and **string**. The default value is **string**.

Validation Rules

When regular expression rules are used for reduction of RADIUS attributes, the following validation rules are applied:

- The protocol field amount must not exceed three fields.
- Each field defined in the fields parameter must have a corresponding [Radius.Field.<field>] section.
- If the separator character is defined, the order of the fields is checked.
- Each field defined in the fields parameter may have a single reduction regular expression rule. If a rule exists, the regular expression validity check is performed.

Configuring the Subscriber ID Example

The following is an example configuration file illustrating how to configure the subscriber ID assignment option. In this example, the User-Name and VPN attributes are assigned to the subscriber ID:

```
[Radius.Subscriber ID]
# Field name
fields=user_name,vpn
# Field separator
# "-" is the separator between the user_name and vpn fields.
field_separator=user_name-vpn
[Radius.Field.user_name]
# RADIUS protocol attribute number
radius_attribute=1
# the type of the attribute (type "integer" or "string")
radius_attribute_type = string
[Radius.Field.vpn]
# Radius protocol attribute number .
radius_attribute = 5
# the type of the attribute (type "integer" or "string")
radius_attribute_type = integer
```

Configuring the Vendor Specific Attribute (VSA) as Subscriber ID Example

The following is an example configuration file illustrating how to configure the subscriber ID assignment option. In this example, the 3GPP_IMSI vendor-specific attribute is assigned to the subscriber ID:

```
[Radius.Subscriber ID]
# Field name
fields=user_name
[Radius.Field.user_name]
# in case of a vendor specific attribute (VSA)
# when the 'radius_attribute' is set to 26
# configuration for 3GPP_IMSI
radius_attribute = 26(10415;1)
# the type of the attribute (type "integer" or "string")
radius_attribute_type = string
```

Configuring Stripping of the Attribute Value Example using Regular Expression Rule

The following is an example configuration file illustrating how to configure the stripping of an attribute value using a regular expression rule:

```
[Radius.Subscriber ID]
# Field name
fields=user_name
# Field manipulation
field_manipulation.user_name=(.*)@.*
[Radius.Field.user_name]
# RADIUS protocol attribute number
radius_attribute=1
# the type of the attribute (type "integer" or "string")
radius_attribute_type = string
```

The above configuration applied on ‘john@some-domain.com’ will extract “john” as a Subscriber ID.

Mapping of RADIUS Attribute to Subscriber Policy

Note The configuration described in this section is optional.

Subscriber policy configuration in the RADIUS Listener can be handled in any of the following ways:

- Extract the data from a RADIUS attribute
- Set a default value for all subscribers that log on via the RADIUS Listener
- Do not set any policy to the subscriber

Extracting Data from a RADIUS Attribute

To define which RADIUS attribute to use for the subscriber policy, configure the **[Radius.Property.Package]** and **[Radius.Field.<field name>]** sections. To define the attribute to be used, configure the following parameters:

- fields

  Defines the RADIUS protocol fields names. When defining multiple fields, use commas between the field names. The field name must not start or end with a space character and it cannot contain an '=' character.

  This parameter has no default value.

  The following is an example of setting this parameter:

  ```
  fields=user_name,ip
  ```

- field_separator

  Defines the character to be used when concatenating several fields.

  The default value is _.

  The following is an example of setting this parameter with the value '-':

  ```
  field_separator=user_name-ip
  ```

- field_manipulation.<field name>=<regular expression>

  The **field_manipulation** parameters define how to manipulate the RADIUS field values.

  The <field name> part of this parameter is one of the fields defined in the **fields** parameter. The <regular expression> part of this parameter is the reduction regular expression to be used on the <field name> value.

  It is possible to define a field_manipulation rule for each name in the field property. This is the default rule if there is a non-configured field manipulation. The following is an example of setting this parameter:

  ```
  field_manipulation.user_name=(.*)@.*
  field_manipulation.ip=(.*)
  ```

- mapping_table.<regExp>=<property-value>

  The **mapping_table** parameters define a conversion table between the result of the attribute value manipulation, the matching rule, and the property value.

  The <regExp>part of this parameter defines the regular expression matching rule. The <property-value>part of this parameter defines the integer result if the regular expression is matched.

  There is no default value for this parameter, but it is possible to set a default value by using the following expression: **mapping_table.^$=<value>**. This value is used if the mapping result is an empty string.

  The following is an example of setting this parameter.

  ```
  mapping_table..*@.*=1
  mapping_table..*=2
  ```

For each field defined by the **fields** parameter, you must also define a **[Radius.Field.<field name>]** section with the following parameters:

- radius_attribute

  Configure the **radius_attribute** parameter with the RADIUS attribute number. Use the following format for Vendor Specific Attributes (VSA): 26(vendor-id;sub-attribute). For example, **26(10415;1)**.

  The default value is **-1**.

- radius_attribute_type

  Configure **radius_attribute_type** parameter according to the RADIUS attribute format.

  Possible values for this parameter are **integer** and **string**. The default value is **string**.

Validation Rules

When regular expression rules are used for reduction of RADIUS attributes, the following validation rules are applied:

- The protocol field amount must not exceed three fields.
- Each field defined in the fields parameter must have a corresponding [Radius.Field.<field>] section.
- The property value in the mapping table is an integer.
- The "=" operator is concatenated to the property value with no space between them.
- A validity test is performed for each regular expression that is used for matching and reduction.

Extracting Data from a RADIUS Attribute Example

The following example is a portion of a configuration file illustrating how to configure the subscriber policy assignment option. In this example, a VSA is assigned to the subscriber policy. It is stripped from its prefix and converted to integer type using a mapping table.

```
[Radius.Property.Package]
# Field name
fields=user_name,vpn
# Field separator
field_separator=user_name@vpn
# Field manipulation
field_manipulation.user_name=(.*)@.*
# Mapping table
mapping_table..*@.*=1
mapping_table.^$=2
mapping_table..*=3
[Radius.Field.user_name]
# Radius protocol attribute number
radius_attribute = 1
# the type of the attribute (type "integer" or "string")
radius_attribute type = string
[Radius.Field.vpn]
# Radius protocol attribute number.
# use the following format for VSAs: 26 vendor-id;sub-attribute)
# for example: 26(9;1)
radius_attribute = 26(9;1)
# the type of the attribute (type "integer" or "string")
radius_attribute_type = string
```

Not Setting Any Policy to the Subscriber

Edit the **[Radius.Property.Package]** section with all remark lines. The number sign ("#") at the beginning of a line signifies a remark line.

Mapping of RADIUS Attribute to Subscriber IP Address

The subscriber IP address is normally based on the Framed-IP-Address attribute; however, it can also be based on a different RADIUS attribute. The default is to use the Framed-IP-Address attribute.

In addition, for environments with IP addresses over VPN, the RADIUS Listener LEG supports the extraction of VPN information from a RADIUS attribute to be used with the extracted IP address.

Note Currently the LEG supports subscriber mappings over VPN only for VPNs that are defined by a VLAN-ID (also referred to as "VPNs of type VLAN").

The following algorithm is applied to handle IP addresses in this LEG:

1. If the user configured an attribute from which to extract the IP, the LEG will look for that attribute in the packet. If the attribute exists, the LEG will use the attribute as the subscriber IP address.
2. If the attribute does not exist or is not configured, the LEG will look for the Framed-Route attributes; several Framed-Route attributes may exist. If any Framed-Route attributes exist, the LEG will use these attributes as the subscriber IP addresses.
3. If there are no Framed-Route attributes, the LEG will look for a Framed-IP-Address attribute and a Framed-IP-Netmask attribute. If a Framed-IP-Address attribute exists, the LEG will use this attribute as the subscriber IP address. If both the Framed-IP-Address and the Framed-IP-Netmask attributes exist, the operation is performed with the IP range represented by the IP address and the IP netmask.
4. Otherwise, the LEG will perform a login without the IP address.

Note The configured attribute can be a regular RADIUS attribute or a VSA. It is possible to encode the attribute as an integer in which case it will be a single IP address. It can also be encoded as a string and will therefore be an IP-Address/IP-Range value: the value must be formatted as A.B.C.D/E or A.B.C.D.

Note The supported format of the Framed-Route attribute is as described in RFC-2865. It must start with a string that starts with the route itself in the format A.B.C.D/E followed by a space. Other values follow the space, but the LEG ignores these other values.

To define which attribute to use for the subscriber IP address, configure the **[Radius.Subscriber IP Address]** and the **[Radius.Field.<field name>]** section. To define the attribute to use, configure the following parameters:

- fields

  Defines the RADIUS protocol field name. Only one field name can be defined.

  The default value is not set.

- vpn_field

  Defines the RADIUS protocol attribute field name to be used as the VPN information related to the IP address.

  The default value is not set.

For the field defined by the **fields** parameter, you must also define a **[Radius.Field.<field name>]** section with the following parameters:

- radius_attribute

  Configure the **radius_attribute** parameter with the RADIUS attribute number. Use the following format for Vendor Specific Attributes (VSA): 26(vendor-id;sub-attribute). For example, **26(10415;1)**.

  The default value is **-1**.

- radius_attribute_type

  Configure **radius_attribute_type** parameter according to the RADIUS attribute format.

  Possible values for this parameter are **integer** and **string**. The default value is **string**.

Configuring Subscriber IP Address Example

The following is an example configuration file illustrating how to configure the subscriber IP assignment option. In the following example, the Framed-IP-Address attribute is used.

```
[Radius.Subscriber IP Address]
# Field name for IP
fields=frame-ip-address
[Radius.Field.frame-ip-address]
# RADIUS protocol attribute number
radius_attribute=8
# the type of the attribute (type "integer" or "string")
# if type is string a mapping table must be supplied
# below.
# (no default)
radius_attribute_type=integer
```

Configuring Subscriber IP Address over VPN Example

The following is an example configuration file illustrating how to configure the subscriber IP assignment option for IP over VPN. In the following example, the Framed-IP-Address attribute is used for the IP address and the cisco-av-pair VSA attribute is used for the VPN information.

```
[Radius.Subscriber IP Address]
# RADIUS protocol field name.
fields=ip
# Radius protocol attribute field name to be used as the
# VPN information related to the IP address.
vpn_field=vpn
[Radius.Field.ip]
# Radius protocol attribute number
radius_attribute = 8
# the type of the attribute (type "integer" or "string")
radius_attribute type = integer
[Radius.Field.vpn]
# Radius protocol attribute number.
# use the following format for VSAs: 26 vendor-id;sub-attribute)
# for example: 26(9;1)
radius_attribute = 26(9;1)
# the type of the attribute (type "integer" or "string")
radius_attribute_type = integer
```

Configuring the NAS Devices

The RADIUS Listener LEG must be configured with the RADIUS clients/NAS devices that transmit RADIUS messages to the LEG, to accept RADIUS messages.

Each **[Radius.NAS.XXX]** section specifies a single Network Access System (NAS), where XXX represents the NAS name.

Step 1 Copy the example Radius.NAS.XXX section that exists in the configuration file

The remarks from the parameters and section header should be removed.

Step 2 Configure a section name of the format **[Radius.NAS.my_name_for_the_NAS]**.

Step 3 Configure the **domain**, **IP_address**, **NAS_identifier**, and **secret** parameters:

- domain

  Set the domain parameter with a valid subscriber domain name.

- IP_address

  Set the IP_address parameter with the NAS IP address with which the RADIUS messages arrive. IP address should be in dotted notation (xxx.xxx.xxx.xxx).

- NAS_identifier

  Set the NAS_identifier parameter with a NAS-ID attribute with which the RADIUS messages are sent.

- secret

  Set the secret parameter with the secret key defined in the NAS for this connection.

Configuring the NAS Devices: Example

This example is a portion of a configuration file illustrating how to configure the NAS:

```
[Radius.NAS.Access134]
# Cisco's subscriber domain name
domain = subscribers
# IP address in dotted notation
IP_address = 202.156.24.100
# name of the NAS that exists in the NAS-ID attribute
NAS_identifier =ACCESS134
# secret string
secret = secret123
```