



Using the Policy Engine

The Cisco Access Registrar (AR) Rule Policy provides an interface to define and configure a policy and to apply the policy to the corresponding access-request packets. Authentication service is based on Realm, Dialed Number Information String (DNIS), and Calling Line ID (CLID).

This chapter contains the following sections:

- [Policies and Rules, page 10-1](#)
- [Script and Attribute Requirements, page 10-2](#)
- [Validation, page 10-3](#)
- [Known Limitations, page 10-3](#)
- [Service Determination, page 10-3](#)
- [Prefix Feature, page 10-8](#)

Policies and Rules

The following is an example set of policies and rules:

```
/Policies
  /SelectPolicy
    Name = SelectPolicy
    Description =
    Grouping = CiscoRealmRule|CiscoCLIDRule

  /CiscoSelectPolicy
    Name = CiscoSelectPolicy
    Description =
    Grouping = CiscoGroupRule

  /CiscoDefaultPolicy
    ...

  /CiscoExecPolicy
    Name = CiscoExecPolicy
    Description =
    Grouping = CiscoExecTimeRule1&CiscoExecSecurityRule|CiscoExecTimeRule2
    ....

/Rules
  /CiscoRealmRule
    Name = CiscoRealmRule
```

```

Description =
Script = ExecRealmRule
/Attributes
    Realm = @cisco.com
    AuthenticationService = jen1-ultra
    Policy = CiscoSelectPolicy

/CiscoCLIDRule
Name = CiscoCLIDRule
Description =
Script = ExecCLIDRule
/Attributes
    ...

/CiscoGroupRule
Name = CiscoGroupRule
Description =
Script = ExecGroupRule
/Attributes
    Group = CiscoExec
    Policy = CiscoExecPolicy
    DefaultPolicy = CiscoDefaultPolicy

/CiscoExecTimeRule1
Name = CiscoExecTimeRule1
Description =
Script = ExecTimeRule
/Attributes
    TimeRange = "mon-fri,06:00-18:00";
    AcceptedProfiles = PPP-USER

/CiscoExecTimeRule2
Name = CiscoExecTimeRule2
Description =
Script = ExecTimeRule
/Attributes
    TimeRange = "sun,18:00-24:00;sat,18:00-24:00";
    AcceptedProfiles = TELNET-USER

/CiscoExecSecurityRule
Name = SecurityRule
Description =
Script = ExecSecurityRule
/Attributes
    TunnelEnforcement = TRUE
    AuthenticationProtocol = CHAP

```

Script and Attribute Requirements

The following requirements exist:

- **/Radius/Policies/SelectPolicy** is the first policy Cisco Access Registrar applies.
- The characters “|” and “&” are reserved as logical operands in a **Grouping** definition; they cannot be included in a **/Radius/Rules** name.



Note The “&” characters has a higher precedence than the “|” in policy grouping expressions.

- A space is not permitted between the logical operands and the rules in a **Grouping** definition.
- The scripts included in the rules must be defined under the **/Radius/Scripts** directory.
- The attributes included in the rules must be defined under the **/Radius/Advanced/Attribute Dictionary** directory.
- The rules included in the policies must be defined under the **/Radius/Rules** directory.

Validation

When policies are configured, the following validations are performed by Cisco Access Registrar:

1. A check is performed to ensure the scripts included in the rules are defined under the **/Radius/Scripts** directory.
2. A check is performed to ensure the attributes included in the rules are defined under the **/Radius/Advanced/Attribute Dictionary** directory.
3. A check is performed to ensure the rules included in the policies are defined under the **/Radius/Rule** directory.

Known Limitations

The following limitations currently exist:

1. **Grouping** expressions are not checked for validity.
2. The use of parentheses to denote precedence is not supported in a **Grouping** definition.
3. A check is not performed to determine whether a policy that is included within another policy is defined under the **/Radius/Policies** directory.

Service Determination

The Cisco Access Registrar server must determine whether to provide local or proxy service by performing Service Determination based on DNIS, Realm, CLID, or a Pattern Match of these parameters

Service determination is set through the Cisco Access Registrar Policy Engine. To make all scripts ready to run, all scripts must be configured and set up through the **aregcmd** command. The following scripts are used for service determination:

- ExecRealmRule
- ExecDNISRule
- ExecCLIDRule

These scripts extract the domain of the username or called-station-id or calling-station-id from the access request packet and compares it with the Realm, DNIS, or CLID set within the rule. Upon finding a match, the scripts will set the services (**Authentication-Service**, **Authorization-Service**, or **Accounting-Service**) into the Environment Dictionary.

In order to invoke this service enhancement, you must add rules and policies. Under the **/Radius/Rules** directory, you set the script that is going to be executed. Under the **/Radius/Policies** directory, you set the combination of rules.

Following is an example for adding a new Realm rule:

```
cd /Radius/Rules
Add RealmRule
cd RealmRule
Set Script ExecRealmRule
cd Attributes
Set Realm @cisco.com
Set Authentication-Service local-users
Set Authorization-Service local-users
```

where **Realm** is the domain filter for user name. If the user-name contains @cisco.com, the **ExecRealmRule** script sets **Authentication-Service** and **Authorization-Service** to local-users. The current release also supports the #cisco.com format.

Besides setting up the rules, you must also set up one or more policies. Policies can be any combination of rules using the *and* (&) and *or* (|) operators. Using the above example, a policy is setup as follows:

```
cd /Radius/Policies
Add SelectPolicy
cd SelectPolicy
Set Grouping RealmRule
```

Inserting, Deleting and Substituting Attributes

This feature supports the RADIUS proxy with the ability to customize attribute filters so that RADIUS attribute value pairs can be inserted/deleted/substituted.

For example, when roaming a packet from ISP A to ISP B, some attribute value (AV) pairs may be substituted (such as IP address) as they may not be valid on B's network. Additionally, B may return some vendor-specific attributes (VSAs) that are not applicable to A's network. Therefore, A will substitute B's VSA value pairs for A's VSAs.

Two configuration points under the **/Radius** directory support this feature: **Translations** and **TranslationGroups**. Under the **/Radius/Translations** directory, any translation to insert, substitute, or translate attributes can be added. The following is a sample configuration under the **/Radius/Translations** directory:

```
cd /Radius/Translations
Add T1
cd T1
Set DeleAttrs Session-Timeout,Called-Station-Id
cd Attributes
Set Calling-Station-Id 18009998888
```

DeleAttrs is the set of attributes to be deleted from the packet. Each attribute is comma separated and no spaces are allowed between attributes.

Under the **/Radius/Translations/T1/Attributes** directory, inserted or translated attribute value pairs can be set. These attribute value pairs are either added to the packet or replaced with the new value.

Under the **/Radius/TranslationGroups** directory, translations can be grouped and applied to certain sets of packets, which are referred to in a rule. The following is a sample configuration under the **/Radius/TranslationGroups** directory:

```
cd /Radius/TranslationGroups
Add CiscoIncoming
cd CiscoIncoming
cd Translations
Set 1 T1
```

The translation group is referenced through the Cisco Access Registrar Policy Engine in the **/Radius/Rules/<RuleName>/Attributes** directory. **Incoming-Translation-Groups** are set to a translation group (for example `CiscoIncoming`) and **Outgoing-Translation-Groups** to another translation group (for example `CiscoOutgoing`).

The following is an example of setting up a rule for a translation group.

```
cd /Radius/Rules
Add CiscoTranslationRule
cd CiscoTranslationRule
cd Attributes
Set Realm @cisco.com
Set Incoming-Translation-Groups CiscoIncoming
Set Outgoing-Translation-Groups CiscoOutgoing
```

The `CiscoTranslationRule` rule must be referred to in the **/Radius/Policies** directory so the Cisco Access Registrar Policy Engine can invoke this rule. If the pattern of **Realm**, **DNIS**, or **CLID** matches the one defined in the rule, the Cisco Access Registrar Policy Engine sets **Incoming-Translation-Groups** to `CiscoIncoming` in the Environment Dictionary. By looking up the definition of `CiscoIncoming`, Cisco Access Registrar applies all of the translations to the incoming packet (before it is proxied to the other server).

When the proxied packet comes back to the RADIUS server, Cisco Access Registrar does a similar process to the outgoing packet.

DNIS, **CLID**, and **Realm** are supported for filtering packets in the current release.

**Note**

Realm in the above example is a filter for packets whose user-name contains `@cisco.com`.

Wildcard Support

Cisco Access Registrar supports limited wildcard functionality. Cisco Access Registrar supports the “*” and “?” wildcard characters. “*” matches any number of characters, including the Null character, and “?” matches any single character, not including the Null character. Currently, wildcards apply to **Realm**, **DNIS**, and **CLID** attributes.

**Note**

All Realms should start with the “@” character. For example, `Realm=@cisco.com`.

The following is an example using the “*” wildcard character:

```

/Radius
  /Rules
    /Rule1
      Name=rule1
      Description =
      ScriptName = ExecRealmRule
      Attributes/
        Authentication-Service = Local-Users
        Authorization-Service = Local-Users
        Realm = @*cisco.com

```

In the above example, when the domain of the user name in an access request matches the @*cisco.com pattern (for example, @us.cisco.com, @eng.cisco.com, and @cisco.com are all good matches), the **ExecRealmRule** script sets **Authentication-Service** and **Authorization-Service** to Local-Users in the environment dictionary.

The following is an example using the “?” wildcard character:

```

/Radius
  /Rules
    /Rule2
      Name = rule2
      Description =
      ScriptName = ExecDNISRule
      Attributes/
        Authentication-Service = Translation-Service
        Authorization-Service = Translation-Service
        DNIS = 1800345987?

```

In the above example, if the **Called-Station-Id** attribute in the packet matches 1800345987? (for example, 18003459876 and 18003459870 are good matches, while 1800345987 is not), the **ExecDNISRule** script sets **Authentication-Service** and **Authorization-Service** to Translation-Service in the environment dictionary.

Cisco Access Registrar also supports both wildcard characters in one pattern. For example, CLID = 180098?87* is valid.

Time of Day Rules

The **ExecTimeRule** script, invoked by the Cisco Access Registrar Policy Engine, applies the **TimeOfDay** rule to the access request packet during the RADIUS server’s incoming packet processing. The **ExecTimeRule** script either rejects or accepts the access request packets based upon the allowable user’s login profiles within a certain time range.

Configuration

The format of the **TimeRange** internal attribute is as follows:

```
TimeRange = dateRange, timeRange [; dateRange, TimeRange]
dateRange = mdayRange | weekdayRange
mdayRange = number [-number]
            number = 1 | 2 | 3 | ... | 31
weekdayRange = weekday [-weekday]
              weekday = sun | mon | tue | wed | thu | fri | sat
timeRange = hh:mm - hh:mm
            hh = 00 | 01 | ... | 23
            mm = 00 | 01 | ... | 59
```

For example:

```
mon-fri,09:00-17:00
mon,09:00-17:00; tue-sat,12:00-13:00
mon,09:00-24:00;tue,00:00-06:00
1-13,10-17:00; 15,00:00-24:00
```

The format of the **AcceptedProfiles** internal attribute is as follows:

```
AcceptedProfiles = userProfile[; userProfile]
```

The following is an example:

```
/Policies
...

/MarketingTODPolicy
  Name = MarketingTODPolicy
  Description =
  Grouping = MarketingTODRule

/EngineeringTODPolicy
  Name = EngineeringTODPolicy
  Description =
  Grouping = EngineeringTODRule

...

/Rules
...

/MarketingTODRule
  Name = MarketingTODRule
  Description =
  Script = ExecTimeRule
  /Attributes
    TimeRange = "mon-fri,8:00-17:00;sat,20:00-23:59;sun,00:00-7:00"
    AcceptedProfile = PPP-users

/EngineeringTODRule
  Name = EngineeringTODRule
  Description =
  Script = ExecTimeRule
  /Attributes
    TimeRange = "sun-sat,00:00-23:59;"
    AcceptedProfiles = PPP-users;Telnet-users

...
```

Notes

Spaces cannot be used in the **TimeRange** or **AcceptedProfiles** attributes.

The lower limit must be less than the upper limit within any specified range.

Validation

Cisco Access Registrar validates the above configuration as follows:

1. Checks whether the **ExecTimeRule** script is defined under the **/Radius/Scripts** directory.
2. Checks whether the **TimeRange** and **AcceptedProfiles** attributes are defined under the **/Radius/Advanced/Attributes** Dictionary.
3. Checks whether the profiles included in the **AcceptedProfiles** attribute are defined under the **/Radius/Profiles** directory.

Known Anomalies

1. Cisco Access Registrar does not check the format of the **TimeRange** attribute.
2. Cisco Access Registrar does not validate the lower limit and the upper limit with the **TimeRange** attribute.

Prefix Feature

The Cisco Access Registrar prefix feature enables you to select a service based on the prefix in the User-Name attribute. Three new attributes and a new script, ExecPrefixRule, have been added to Cisco AR.

The prefix feature supports wildcard matching and multi-valued attributes by which multiple prefixes can be configured under the same rule.

New Attributes

Three new attributes have been added to Cisco AR to support the prefix feature:

Prefix—List of valid prefixes

Delimiters—List of valid delimiters

StripPrefix—Option to strip or not to strip the prefix from the User-Name. If you do not configure this attribute, the default (Yes) is to strip the prefix from the User-Name. Allowed values are Yes and No.

Example Configuration

The following example configuration is based on a new rule created to support the prefix feature. The new rule should be set under a Policy.

In the example configuration, if **cisco/bob@cisco.com** is the User-Name attribute, the ExecPrefixRule script sets the Authentication-Service to **cisco** and User-Name to **cisco/bob@cisco.com** because the StripPrefix attribute is configured **No**.

If the StripPrefix attribute is not configured, or if it is configured to **Yes**, the prefix will be stripped from the UserName, and UserName will become **bob@cisco.com**. The following steps provide an example configuration.

Step 1 Using **aregcmd**, navigate to **//localhost/Radius/Rules**.

```
[ //localhost/Radius/Rules ]
  Entries 0 to 0 from 0 total entries
  Current filter: <all>
```

Step 2 Add a new rule.

```
-->add rule1
```

```
Added rule1
```

Step 3 Change directory to that rule, and set the rule's script to ExecPrefixRule.

```
--> cd rule1
```

```
[ //localhost/Radius/Rules/rule1 ]
  Name = rule1
  Description =
  Script~ =
  Attributes/
```

```
--> set Script ExecPrefixRule
```

```
Set Script ExecPrefixRule
```

Step 4 Change directory to **Attributes**, and set the attributes to the following:

```
--> cd Attributes
```

```
[ //localhost/Radius/Rules/rule1/Attributes ]
```

```
--> set Authentication-Service cisco
```

```
Set Authentication-Service cisco
```

```
--> set Prefix "cisco" "ibm"
```

```
Set Prefix cisco ibm
```

```
--> set Delimiters @#%&/
```

```
Set Delimiters @#%&/
```

```
--> set StripPrefix No
```

```
Set StripPrefix No
```

```
--> ls
```

```
[ //localhost/Radius/Rules/rule1/Attributes ]
```

```
Authentication-Service = cisco
Delimiters = @#%&/
Prefix = cisco
Prefix = ibm
StripPrefix = No
```

Step 5 Save the configuration changes.

```
-->save
```

```
Validating //localhost...
Saving //localhost...
```

```
->
```