Using FastRules to Process Packet Flow

While using rule policy engine and scripting points to process packet flow, you need to be familiar with programming languages, and create scripts to attach them to the Prime Access Registrar configuration. FastRules concept is an easier and efficient alternative to rule policy engine and scripting points.

FastRules provides a mechanism to easily choose the right authentication, authorization, accounting, and query service(s), drop, reject, or break flows, run a script, choose a session manager and/or a chain of fast rules required for processing a packet.

FastRules has the following capabilities:

- Provides maximum flexibility and ease in matching information in the incoming packets for choosing the appropriate service to apply

- Provides an option to match values in AVPs based on value ranges, exact match, and simple string comparisons using regex

- Provides easy and efficient alternative to rule/policy engine and scripting points for most common use cases—reduces the use of external scripts to choose an appropriate service

Figure 19-1 describes the workflow for FastRules.

FastRules also provides an option to use Internal Scripts as part of the workflow. Internal scripts allow you to add, modify, or delete attributes in the request, response, and environment dictionaries for RADIUS, Diameter, and TACACS+. For more information about internal scripts, see Internal Scripts, page 11-14.

Figure 19-1   FastRules Workflow
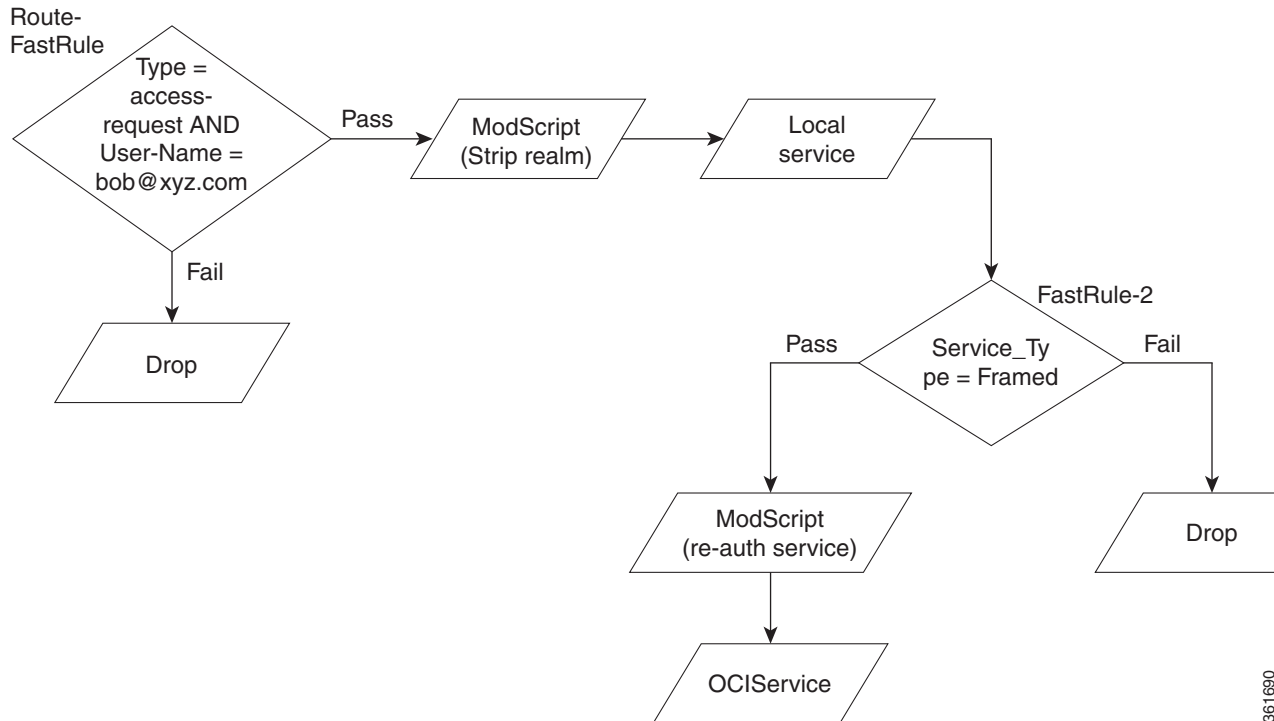
Configuring FastRules

FastRules can be configured either through the GUI or through a CLI. For more information on configuring fast rules through the GUI, see FastRules, page 3-53.

A sample CLI for configuring fast rules is given below:

```
[ //localhost/Radius/Fastrules]
    Ruledefintions\
    radroot\
        Name = radroot
        Desc =
        Protocol = Radius
        Condition = A1 AND (A2 OR A3)
        Success = Authen(mod1)-->fr2-->Author(oci)
        Failure =
        Attributes\
        A1\
            Name = A1
            Attribute = Calling-station-id
            Value = 1.2.3.4
        A2\
            Name = A2
            Attribute = User-name
            Value = bob
        A3\
            Name = A3
            Attribute = User-name
            Value = BEGINS_WITH(jane)
    fr1\
        Name = fr1
```

```
                    Desc =
                    Protocol = Radius
                    Condition = Attri1
                    Success = Authen(proxy)
                    Failure = DROP
                    Attributes\
                    Attri1\
                        Name = Attr1
                        Attribute = Calling-station-id
                        Value = 2.3.4.5
            fr2\
                    Name = fr2
                    Desc =
                    Protocol = Radius
                    Condition = Attribute1
                    Success = Authen(local)
                    Failure =
                    Attributes\
                        Attribute1\
                            Name = Attribute1
                            Attribute = Status
                            Value = Failure
        .
            Order\
                    Radius\
                        1. radroot
                        2. fr1
                    Diameter\
                        1.
                    Tacacs\
                        1.

        Services\
            mod1\
            Name = mod1
            Desc =
            Type = mod
            Service = ldap
            Dictionary\
                +env:Calling = req:Calling-Station-Id
                -req:Calling-Station-Id = ""
                +req:Calling-Station-Id = "5.6.7.8"
                +res:Calling-Station-Id = env:Calling
              #res:Called-Station-Id = "123456789"
```