Diameter

Diameter is a networking protocol which is derived from RADIUS protocol. It is considered to be the next generation Authentication, Authorization, and Accounting (AAA) protocol. This is the other core protocol used in the IP Multimedia Subsystem (IMS) architecture for IMS Entities to exchange AAA related information. Cisco Prime Access Registrar (Prime Access Registrar) supports Diameter Applications based on the Diameter Base Protocol defined in RFC 6733.

Diameter is composed of a base protocol and a set of applications which allows it to extend its services to new access technologies. The base protocol provides basic mechanisms for reliable transport, message delivery, and error handling. Each application is defined by an application identifier and associated with commands. Each command is defined with mandatory Attribute Value Pairs (AVPs) and non-mandatory AVPs including vendor-specific AVPs.

The base protocol must be used in conjunction with a Diameter application. Each application relies on the services of the base protocol to support a specific type of network access.

The following is the list of applications supported by Prime Access Registrar:

- Diameter Network Access Server Application (NASREQ, RFC 4005)
- Diameter Base Accounting (RFC 6733)
- Diameter Extensible Authentication Protocol (EAP) Application (RFC 4072)

This chapter contains the following sections:

Diameter with EAP Support

The Extensible Authentication Protocol (EAP), is an authentication framework which supports multiple authentication mechanisms. EAP may be used on dedicated links, switched circuits, and wired as well as wireless links. For more information on EAP support in Prime Access Registrar, see Chapter 5, "Extensible Authentication Protocols."

Prime Access Registrar supports Diameter EAP application that carries EAP packets between a Network Access Server (NAS) working as an EAP Authenticator and a back-end authentication server. The Diameter EAP application is based on the Diameter Network Access Server Application [NASREQ] and is intended for environments similar to NASREQ.

In the Diameter EAP application, authentication occurs between the EAP client and its home Diameter server. This end-to-end authentication reduces the possibility for fraudulent authentication, such as replay and man-in-the-middle attacks. End-to-end authentication also provides a possibility for mutual authentication, which is not possible with PAP and CHAP in a roaming PPP environment.

This topic contains the following sections:

Advertising Application Support

Diameter nodes conforming to this specification must advertise support by including the Diameter EAP Application ID value of 5 in the Auth-Application-Id AVP of the Capabilities-Exchange-Request and Capabilities-Exchange-Answer command [BASE].

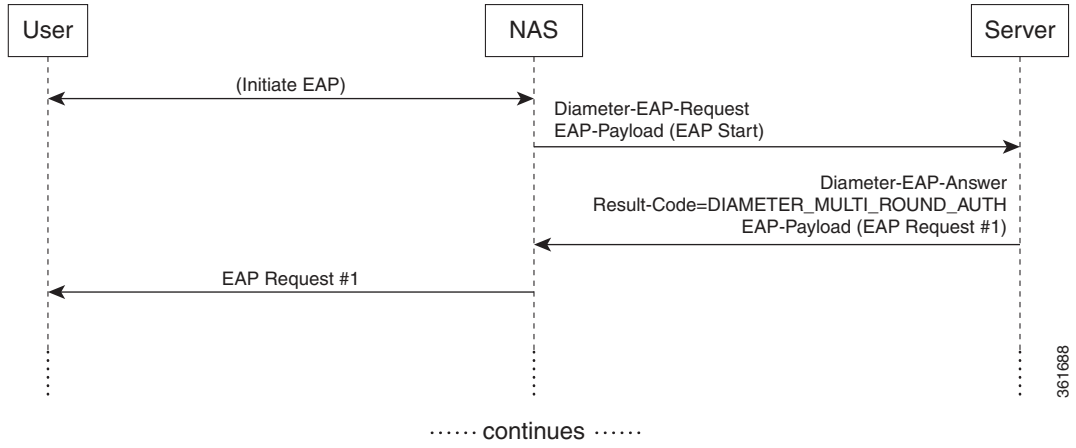
If the NAS receives a response with the Result-Code set to DIAMETER_APPLICATION_UNSUPPORTED [BASE], it indicates that the Diameter server in the home realm does not support EAP. If possible, the access device may attempt to negotiate another authentication protocol, such as PAP or CHAP. An access device must be cautious when determining whether a less secure authentication protocol will be used, since this could result from a downgrade attack.

Diameter EAP Conversation Flow

The EAP conversation between the authenticating peer and the access device begins with the initiation of EAP within a link layer, such as PPP [RFC1661] or IEEE 802.11i [IEEE-802.11i]. Once EAP has been initiated, the access device will typically send a Diameter-EAP- Request message with an empty EAP-Payload AVP to the Diameter server, signifying an EAP-Start. Prime Access Registrar routes the message to the Diameter EAP service through the rules and policy engine (and/or client, server and vendor scripting point) through which the packet is processed. The Diameter EAP Service forms a Diameter-EAP-Answer message containing an EAP-Payload AVP that includes an encapsulated EAP packet. The Result-Code AVP in the message will be set to DIAMETER_MULTI_ROUND_AUTH, signifying that a subsequent request is expected.

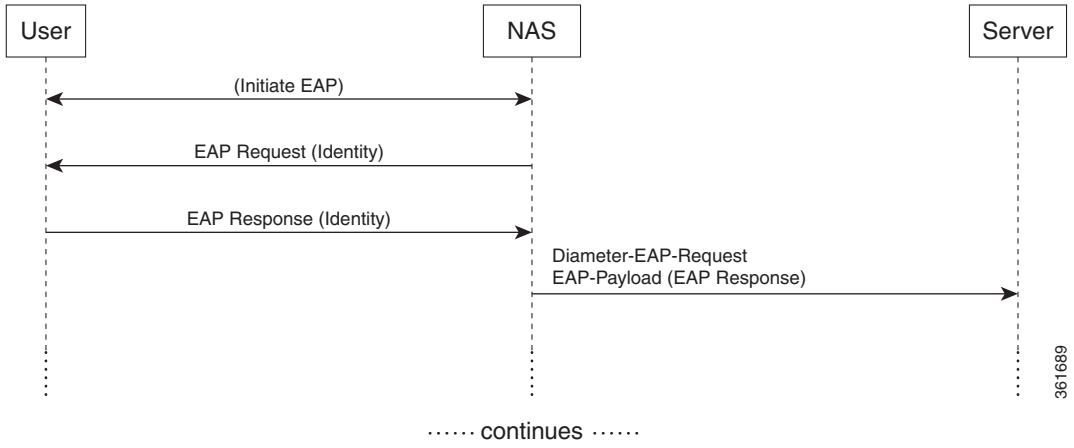
Figure 4-1 describes the Diameter EAP request flow.

Figure 4-1 Diameter EAP Request Flow



The access device issues the EAP-Request/Identity message to the EAP client, and forwards the EAP-Response/Identity packet, encapsulated within the EAP-Payload AVP, as a Diameter-EAP-Request to Prime Access Registrar as shown in Figure 4-2. This reduces the number of Diameter message round trips.

Figure 4-2 Diameter EAP Response Flow



The conversation continues until the Diameter server sends a Diameter-EAP-Answer with a Result-Code AVP indicating success or failure, and an optional EAP-Payload. The Result-Code AVP is used by the access device to determine whether service is to be provided to the EAP client or not. The access device must not rely on the contents of the optional EAP-Payload to determine whether service is to be provided or not.

Diameter Server Startup Log

When Prime Access Registrar starts, Diameter server also starts.

The log file shows the following:

```
09/30/2013 6:38:47.419 name/radius/1 Info Server 0 Diameter Server Started
09/30/2013 6:38:47.437 name/radius/1 Info Protocol 0  Starting diameter core
```

```
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0            Product : Cisco Prime
Access Registrar
09/30/2013 6:38:47.447 name/radius/1 Info Protocol 0             Version : 6
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0           Vendor Id : 0
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0    Auth Application : 0
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0    Auth Application : 1
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0    Acct Application : 3
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0          Dictionary :
/cisco-ar/conf/diadictionary.xml
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0            Identity :
10.81.79.43
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0               Realm : abc.com
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0          TCP Listen : 3868
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0         SCTP Listen : 3868
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0   Watch-Dog timeout : 500
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0           Use IPv6 : 0
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0 Re-transmission Int : 8
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0    Max Re-trans Int : 3
09/30/2013  6:38:47.447 name/radius/1 Info Protocol 0    Recv Buffer Size : 20480
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0      Hostnames Used :
10.81.79.43
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0  Dumping Peer Table
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0       Expire Time 1
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0     Peer : Host = 10.77.240.54,
Port = 3868, AdvertiseHostName =  , AdvertisedRealm =  , TLS = 0
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0     Peer : Host = 10.77.240.53,
Port = 3868, AdvertiseHostName =  , AdvertisedRealm=  , TLS = 0
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0   Dumping Route Table
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0             Exp Time : 0
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0               Route  : Realm =
dia.com, Action = 2, Redirect-Usage = 0
09/30/2013  6:38:47.448 name/radius/1 Info Protocol 0
Application Id=1, Vendor=0
09/30/2013  6:38:47.449 name/radius/1 Info Protocol 0                          Server
= 10.77.240.53, metric = 2
09/30/2013  6:38:47.449 name/radius/1 Info Protocol 0  Auth Stateful Auth : stateful
09/30/2013  6:38:47.449 name/radius/1 Info Protocol 0     Auth Session(T) : 30
09/30/2013  6:38:47.449 name/radius/1 Info Protocol 0    Auth Lifetime(T) : 360
09/30/2013  6:38:47.449 name/radius/1 Info Protocol 0       Auth Grace(T) : 30
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0       Auth Abort(T) : 20
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0     Acct Session(T) : 30
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0    Acct Interim Int : 5
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0      Acct Real-Time : 0
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0           Debug Log : enabled
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0           Trace Log : enabled
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0            Info Log : enabled
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0         Console Log : enabled
09/30/2013  6:38:47.450 name/radius/1 Info Protocol 0          Syslog Log : disabled
```

Diameter Stack Level Messages

The following are the stack level messages that are exchanged between the diameter peers:

- Capabilities Exchange Message
- Watchdog Message
- Disconnect Message

Capabilities Exchange Message

When Diameter peers establish a transport connection to Prime Access Registrar, they will exchange the Capabilities Exchange messages. This message allows the discovery of a peer's identity and its capabilities (protocol version number, supported Diameter applications, security mechanisms, etc.)

The log file shows the following:

```
05/14/2015  5:36:19.646 name/radius/1 Info Server 0 Starting Server
05/14/2015  5:36:19.707 name/radius/1 Info Server 0 RollingEncryption using new key 17
and aging key 18
05/14/2015  5:36:19.732 name/radius/1 Info Server 0 RollingEncryption using new key 17
and aging key 18
05/14/2015  5:36:19.852 name/radius/1 Info Server 0 Device-Watchdog-Request thread
activated for peer 10.81.79.60
05/14/2015  5:36:19.852 name/radius/1 Info System 0  Connecting to 10.81.79.60:4000
...
05/14/2015  5:36:19.852 name/radius/1 Info System 0 Local socket bind on 10.81.79.81:0
05/14/2015  5:36:20.852 name/radius/1 Info Protocol 0 Connected to RemoteServer b2
05/14/2015  5:36:20.853 name/radius/1 Info Server 0 Initiating CER to 10.81.79.60...
05/14/2015  5:36:20.853 name/radius/1 Info Server 0 Received CEA from the
peer(10.81.79.60), IP: 10.77.240.41
05/14/2015  5:36:20.854 name/radius/1 Info Server 0 Capabilities are successfully
exchanged with 10.81.79.60
05/14/2015  5:36:21.053 name/radius/1 Info Server 0 Sticky Sessions BGwrite thread
activated
05/14/2015  5:36:21.053 name/radius/1 Info Server 0 Sticky Session Count BG thread
activated.
05/14/2015  5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
1812 (RADIUS Access)
05/14/2015  5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
1813 (RADIUS Accounting)
05/14/2015  5:36:21.058 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
49 (TACACS+)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
3868 (Diameter-TCP)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 127.0.0.1, port
3868 (Diameter-SCTP)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 1812 (RADIUS Access)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 1813 (RADIUS Accounting)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 49 (TACACS+)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 3868 (Diameter-TCP)
05/14/2015  5:36:21.059 name/radius/1 Info Server 0 Starting Interface 10.81.79.81,
port 3868 (Diameter-SCTP)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 1812 (RADIUS Access)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 1813 (RADIUS Accounting)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 49 (TACACS+)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 3868 (Diameter-TCP)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting IPv6 Interface
2001:420:27c1:421:250:56ff:fe99:9e20, port 3868 (Diameter-SCTP)
05/14/2015  5:36:21.060 name/radius/1 Error Configuration 0 Interface
fe80::250:56ff:fe99:9e20: af_bind() to port 1812 failed with -2147418090
05/14/2015  5:36:21.060 name/radius/1 Error Server 0 Failed to start IPv6 Interface
fe80::250:56ff:fe99:9e20, port 1812 (RADIUS Access)
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Starting Replication Manager
```

```
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 Replication Disabled
05/14/2015  5:36:21.060 name/radius/1 Info Server 0 SNMP is disabled
05/14/2015  5:36:21.061 name/radius/1 Info Server 0 Memory limit for Radius process is
activated
05/14/2015  5:36:21.061 name/radius/1 Info Server 0 Server Started Successfully (pid:
728)
```

Watchdog Message

The Device-Watchdog-Request and Device-Watchdog-Answer messages are used to proactively detect transport failures. Device Watchdog message time interval is configurable in Prime Access Registrar.

Disconnect Message

Disconnect messages are initiated when Diameter peers lose transport connection to Prime Access Registrar.

Configuring Authentication and Authorization for Diameter

This section describes how to configure Prime Access Registrar to perform authentication and authorization and how to configure a local service and userlist.

See for more information on Diameter client properties.

This section contains the following topics:

- Configuring Local Authentication and Authorization
- Configuring External Authentication Service

Configuring Local Authentication and Authorization

In Diameter, an AA-Request packet is a request for authentication and authorization. Authentication checks username and password credentials, while authorization typically involves returning the correct information to allow the service a user is authorized to have. Prime Access Registrar performs AA and returns the appropriate Diameter attributes in an AA-Answer packet.

For adding a Diameter peer in Prime Access Registrar, configure a new entry in the clients (including Policy and Charging Rules Functions (PCRF), Home Subscriber Servers (HSS), Mobility Management Entities (MME), Online Charging Systems (OCS), and others) and remote server object.

The following shows an example configuration for adding a Diameter peer (NAS/Client) in Prime Access Registrar.

```
[ //localhost/Radius/Clients/70dia ]
    Name = 70dia
    Description =
    Protocol = diameter
    HostName = 10.81.79.241
    PeerPort = 3868
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    AdvertisedHostName =
```

```
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
WatchDogTimeout = 500
SCTP-Enabled = FALSE
TLS-Enabled = FALSE


[ //localhost/Radius/Services/diaservice ]
    Name = diaservice
    Description =
    Type = diameter
    IncomingScript~ =
    OutgoingScript~ =
    EnableSticky = FALSE
    MultiplePeersPolicy = Failover
    PeerTimeOutPolicy = FailOver
    DiaRemoteServers/
        Entries 1 to 1 from 1 total entries
        Current filter: <all>

        65/
            Name = 65
            Metric = 2
            Weight = 0
            IsActive = TRUE
```

Note You should restart the Prime Access Registrar server if you change any Diameter specific configuration.

See and for more details.

Configuring a Local Service and UserList

See for more information on how to configure a local service and user list.

The following messages are logged in the trace file at the time of authenticating a valid user:

```
05/14/2015  5:41:47.574: P734: Packet received from 10.81.79.81
05/14/2015  5:41:47.574: P734: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015  5:41:47.574: P734: Using Client: vm050
05/14/2015  5:41:47.574: P734: Packet successfully added
05/14/2015  5:41:47.574: P734: Trace of Diameter Packet
05/14/2015  5:41:47.574: P734:    Destination-Realm = cisco.com
05/14/2015  5:41:47.574: P734:    User-Name = bob
05/14/2015  5:41:47.574: P734:    User-Password = <encrypted>
05/14/2015  5:41:47.574: P734:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  5:41:47.574: P734:    Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015  5:41:47.574: P734:    Session-Id = .;2096298391;2
05/14/2015  5:41:47.574: P734:    Auth-Application-Id = 1
05/14/2015  5:41:47.574: P734:    Origin-Realm = xyz.com
05/14/2015  5:41:47.574: P734: Tracing the packet after running the rules and policies
05/14/2015  5:41:47.574: P734: Using Client: vm050
05/14/2015  5:41:47.574: P734:  FastRule Engine called for packet
05/14/2015  5:41:47.574: P734: Fastrule return = 0
05/14/2015  5:41:47.574: P734: Authorizing with Service local-users
05/14/2015  5:41:47.574: P734: Getting User bob's UserRecord from UserList Default
05/14/2015  5:41:47.575: P734: User bob is part of UserGroup PPP-users
05/14/2015  5:41:47.575: P734: Merging UserGroup PPP-users's BaseProfiles into
response dictionary
```

```
05/14/2015  5:41:47.575: P734: Merging UserGroup PPP-users's Attributes into response
Dictionary
05/14/2015  5:41:47.575: P734: Merging attributes into the Response Dictionary:
05/14/2015  5:41:47.575: P734: Trace of Diameter Packet
05/14/2015  5:41:47.575: P734:    User-Name = bob
05/14/2015  5:41:47.575: P734:    Result-Code = Diameter-Success
05/14/2015  5:41:47.575: P734:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  5:41:47.575: P734:    Origin-Host = 10.81.79.81
05/14/2015  5:41:47.575: P734:    Session-Id = .;2096298391;2
05/14/2015  5:41:47.575: P734:    Auth-Application-Id = 1
05/14/2015  5:41:47.575: P734:    Origin-Realm = cisco.com
05/14/2015  5:41:47.575: P734: Sending response to ar-lnx-vm050.cisco.com
05/14/2015  5:41:47.575: P734: Packet successfully removed
05/14/2015  5:41:47.575: P734: Packet Deleted
```

The following messages are logged in the trace file at the time of authenticating an invalid user:

```
05/14/2015  5:45:29.478: P831: Packet received from 10.81.79.81
05/14/2015  5:45:29.478: P831: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015  5:45:29.478: P831: Using Client: vm050
05/14/2015  5:45:29.478: P831: Packet successfully added
05/14/2015  5:45:29.478: P831: Trace of Diameter Packet
05/14/2015  5:45:29.478: P831:    Destination-Realm = cisco.com
05/14/2015  5:45:29.478: P831:    User-Name = user.1
05/14/2015  5:45:29.478: P831:    User-Password = <encrypted>
05/14/2015  5:45:29.478: P831:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  5:45:29.479: P831:    Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015  5:45:29.479: P831:    Session-Id = .;2096298391;3
05/14/2015  5:45:29.479: P831:    Auth-Application-Id = 1
05/14/2015  5:45:29.479: P831:    Origin-Realm = xyz.com
05/14/2015  5:45:29.479: P831: Tracing the packet after running the rules and policies
05/14/2015  5:45:29.479: P831: Using Client: vm050
05/14/2015  5:45:29.479: P831:  FastRule Engine called for packet
05/14/2015  5:45:29.479: P831: Fastrule return = 0
05/14/2015  5:45:29.479: P831: Authorizing with Service local-users
05/14/2015  5:45:29.479: P831: Getting User user.1's UserRecord from UserList Default
05/14/2015  5:45:29.479: P831: No UserRecord found for User user.1 in UserList
Default, but none _required_ for Authorization.
05/14/2015  5:45:29.479: P831: Trace of Diameter Packet
05/14/2015  5:45:29.479: P831:    User-Name = user.1
05/14/2015  5:45:29.479: P831:    Result-Code = Diameter-Authentication-Rejected
05/14/2015  5:45:29.479: P831:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  5:45:29.479: P831:    Origin-Host = 10.81.79.81
05/14/2015  5:45:29.479: P831:    Session-Id = .;2096298391;3
05/14/2015  5:45:29.479: P831:    Auth-Application-Id = 1
05/14/2015  5:45:29.479: P831:    Origin-Realm = cisco.com
05/14/2015  5:45:29.479: P831: Sending response to ar-lnx-vm050.cisco.com
05/14/2015  5:45:29.479: P831: Packet successfully removed
05/14/2015  5:45:29.480: P831: Packet Deleted
```

Configuring External Authentication Service

See for more information on how to configure external authentication service.

Configuring the Diameter Application in Prime Access Registrar

For proxying a diameter application message in Prime Access Registrar, ensure that you do the following:

Configuring the Transport Management Properties

You need to log into the aregcmd using the CLI interface and configure the Transport Management properties in the **Radius/Advanced/Diameter/**.

```
[ //localhost/Radius/Advanced/Diameter ]
    IsDiameterEnabled = TRUE
    General/
        Product = CPAR
        Version = 7.2.0.0
        AuthApplicationIdList = 1
        AcctApplicationIdList = 3
    TransportManagement/
        Identity = 10.77.240.69
        Realm = abc.com
        WatchdogTimeout = 500
        ValidateIncomingMessages = FALSE
        ValidateOutgoingMessages = TRUE
        MaximumNumberofDiameterPackets = 8194
        ReserveDiameterPacketPool = 0
        DiameterPacketSize = 2048
        AdvertisedHostName/
        SCTPOptions/
            MaxInitRetry = 3
            MaxInboundStream = 4
            MaxOutboundstream = 5
            EnableHeartbeat = FALSE
            HeartbeatInterval = 500
```

You need to set the Identity and AdvertisedHostName properties to IP Address or hostname of the machine in which Prime Access Registrar is installed.

```
--> set Identity 10.77.240.69
Set Identity 10.77.240.69

--> cd AdvertisedHostName
set 1 10.77.240.69
Set the Realm in which Cisco Prime Access Registrar server is present.
--> set Realm cisco.com
Set Realm cisco.com

Save the configuration

--> save

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

ls
[ //localhost/Radius/Advanced/Diameter/TransportManagement ]
    Identity = 10.77.240.69
    Realm = cisco.com
    WatchdogTimeout = 500
    ValidateIncomingMessages = FALSE
```

```
ValidateOutgoingMessages = TRUE
MaximumNumberofDiameterPackets = 8194
ReserveDiameterPacketPool = 0
DiameterPacketSize = 2048
AdvertisedHostName/
    1. 10.77.240.69
SCTPOptions/
    MaxInitRetry = 3
    MaxInboundStream = 4
    MaxOutboundstream = 5
    EnableHeartbeat = FALSE
    HeartbeatInterval = 500
```

The description for these properties is available at:

http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/5.1/user/guide/objects.html#wp1145662

Note Prime Access Registrar can only listen to one port for diameter connections. In the above configuration, the port number is 3868. All of the diameter clients must use this port number to communicate with the Prime Access Registrar.

Registering Applications IDs

You need to register the applications IDs for which Prime Access Registrar needs to route the Diameter Messages.

Registering the Gy application to a diameter stack

To register the Gy application to a diameter stack,

Step 1 Move to the **//localhost/Radius/Advanced/Diameter/General** directory.

```
[ //localhost/Radius/Advanced/Diameter ]
    IsDiameterEnabled = TRUE
    General/
    TransportManagement/


--> cd General/

[ //localhost/Radius/Advanced/Diameter/General ]
Product = Cisco Prime Access Registrar
Version = 7.2.0.0
AuthApplicationIdList = 1
AcctApplicationIdList =
```

For description of these properties, see .

Step 2 Set the AuthApplicationIdList to list of colon separated values of Application Ids.

```
--> set AuthApplicationIdList "4"

Set AuthApplicationIdList 4
```

Configuring the Diameter Peers

You need to configure the Diameter Peers such as clients and servers in the **/radius/clients** and **/radius/remoteservers** directories. The following is an example for configuring a Diameter client:

```
[ //localhost/Radius/Clients/70dia ]
    Name = 70dia
    Description =
    Protocol = diameter
    HostName = 10.81.79.241
    PeerPort = 3868
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    AdvertisedHostName =
    AdvertisedRealm =
    InitialTimeout = 1000
    MaxIncomingRequestRate = 0
    WatchDogTimeout = 500
    SCTP-Enabled = FALSE
    TLS-Enabled = FALSE
```

The following is an example for configuring a Diameter remote server:

```
[ //localhost/Radius/RemoteServers/lap ]
    Name = lap
    Description =
    Protocol = diameter
    HostName = 10.77.144.34
    Port = 3868
    DestinationRealm = cisco.com
    ReactivateTimerInterval = 300000
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 3
    MaxTPSLimit = 0
    MaxSessionLimit = 0
    InitialTimeout = 2000
    LimitOutstandingRequests = FALSE
    MaxPendingPackets = 0
    MaxOutstandingRequests = 0
    DWatchDogTimeout = 2500
    SCTP-Enabled = FALSE
    TLS-Enabled = FALSE
    AdvertiseHostName =
    AdvertiseRealm =
```

For description of these properties, see .

Note In order to resolve the hostnames and get the IP addresses, the Prime Access Registrar should either be configured with a DNS server IP, or the client's hostnames and IP addresses should be included in the /etc/hosts file.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 Prime Access Registrar localhost.localdomain localhost
172.16.29.7 GGSN-Gy
::1 localhost6.localdomain6 localhost6
```

Configure the Diameter Service

To configure the Diameter Service to route the Diameter Messages,

Step 1 Add a Service of type diameter in /Radius/Services/.

```
--> cd  /Radius/Services/
--> add  dia-proxy

Added dia-proxy


--> cd dia-proxy

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type =

--> set Type diameter

Set Type diameter
```

Step 2 Configure the sticky properties.

```
--> set EnableSticky TRUE

Set EnableSticky TRUE

--> ls

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type = diameter
    IncomingScript~ =
    OutgoingScript~ =
    EnableSticky = TRUE
    StickySessionKey =
    StickyCreationCmdList =
    StickyDeletionCmdList =
    MultiplePeersPolicy = Failover
    PeerTimeOutPolicy = FailOver
    DiaRemoteServers/

--> set StickySessionKey Session-Id#1

Set StickySessionKey Session-Id#1

--> set StickyCreationCmdList 265

Set StickyCreationCmdList 265

--> set StickyDeletionCmdList 275

Set StickyDeletionCmdList 275

--> set MultiplePeersPolicy RoundRobin
```

```
Set MultiplePeersPolicy RoundRobin

--> ls

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type = diameter
    IncomingScript~ =
    OutgoingScript~ =
    EnableSticky = TRUE
    StickySessionKey = Session-Id#1
    StickyCreationCmdList = 265
    StickyDeletionCmdList = 275
    MultiplePeersPolicy = RoundRobin
    PeerTimeOutPolicy = FailOver
    DiaRemoteServers/
```

Step 3 Add the peers to which Prime Access Registrar needs to load balance the diameter messages.

```
[ //localhost/Radius/RemoteServers/dia1 ]
    Name = dia1
    Description =
    Protocol = diameter
    HostName = 192.168.30.88
    Port = 3868
    DestinationRealm =
    ReactivateTimerInterval = 300000
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 3
    MaxTPSLimit = 0
    MaxSessionLimit = 0
    InitialTimeout = 2000
    LimitOutstandingRequests = FALSE
    MaxPendingPackets = 0
    MaxOutstandingRequests = 0
    DWatchDogTimeout = 2500
    SCTP-Enabled = FALSE
    TLS-Enabled = FALSE
    AdvertiseHostName =
    AdvertiseRealm =
[ //localhost/Radius/RemoteServers/dia2 ]
    Name = dia2
    Description =
    Protocol = diameter
    HostName =
    Port = 0
    DestinationRealm =
    ReactivateTimerInterval = 300000
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 3
    MaxTPSLimit = 0
    MaxSessionLimit = 0
    InitialTimeout = 2000
    LimitOutstandingRequests = FALSE
    MaxPendingPackets = 0
    MaxOutstandingRequests = 0
    DWatchDogTimeout = 2500
    SCTP-Enabled = FALSE
    TLS-Enabled = FALSE
```

```
    AdvertiseHostName =
    AdvertiseRealm =

--> cd diaRemoteServers/

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers ]
    Entries 0 to 0 from 0 total entries
    Current filter: <all>


--> add dia1

Added dia1

--> cd dia1/

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers/dia1 ]
    Name = dia1
    Metric = 2
    Weight = 0
    IsActive = TRUE

--> cd ..

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers ]
    Entries 1 to 1 from 1 total entries
    Current filter: <all>

    dia1/

--> add dia2

Added dia2

--> cd dia2

[ //localhost/Radius/Services/dia-proxy/DiaRemoteServers/dia2 ]
    Name = dia2
    Metric = 3
    Weight = 0
    IsActive = TRUE
```

Step 4 Save the configuration details.

```
--> save
Validating //localhost...
Saving //localhost...
```

Step 5 Set DefaultAuthenticationService and DefaultAuthorizationService in /Radius directory.

```
--> set DefaultAuthenticationService dia-proxy

     Set DefaultAuthenticationService dia-proxy

--> set DefaultAuthorizationService dia-proxy

     Set DefaultAuthorizationService dia-proxy

--> save
Validating //localhost...
Saving //localhost...

--> exit
Logging out of localhost...
```

Step 6 Restart thePrime Access Registrar server.

```
/cisco-ar/bin/arserver restart
```

The following illustrates the diameter proxy service configuration which load balances the diameter messages to the remote peers.

```
[ /Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type = diameter
    IncomingScript~ =
    OutgoingScript~ =
    EnableSticky = TRUE
    StickySessionKey = Session-Id#1
    StickyCreationCmdList = 265
    StickyDeletionCmdList = 275
    MultiplePeersPolicy = RoundRobin
    PeerTimeOutPolicy = FailOver
    DiaRemoteServers/
        Entries 1 to 2 from 2 total entries
        Current filter: <all>

        dia1/
            Name = dia1
            Metric = 2
            Weight = 0
            IsActive = TRUE
        dia2/
            Name = dia2
            Metric = 3
            Weight = 0
            IsActive = TRUE
```

For description of these properties, see .

Group-Based Load Balancing in Diameter Proxy Server Configuration

Prime Access Registrar allows you to create two or more groups of Diameter remote servers in a Diameter proxy service configuration. Each of these groups will have a unique set of remote servers, i.e. no two groups will share the same remote server.

The traffic between each of these groups is load-balanced in failover mode; while traffic between remote servers within the same group is load-balanced based on round-robin or failover mode depending on the existing Diameter configuration. The priority of each of the groups is set with the help of metrics.

The workflow for group-based load balancing is as given below:

1. Traffic from Prime Access Registrar to remote server, via Diameter proxy service, is directed through the first group, till Prime Access Registrar has active communication channel with at least one remote server belonging to the first group.
2. When Prime Access Registrar loses connectivity with all the remote servers in the first group, it directs the rest of the Diameter traffic towards remote servers belonging to the second group.
3. Within a group, the load-balancing logic is chosen based on the configuration:
    a. If the load-balancing logic is configured to be round-robin, the traffic is distributed across all the active remote servers.

b. If the load-balancing logic is configured to be failover, the traffic is directed towards first priority remote server. When Prime Access Registrar loses connectivity with the first priority remote server, it directs the subsequent traffic towards the second priority remote server. The priority of the Diameter remote servers, in case of failover logic, is set with the help of metrics.

For more information about Diameter server group parameters, see GroupServers, page 2-22.

Following is a sample configuration of the Diameter group server:

```
[ //localhost/Radius/GroupServers ]
    Entries 1 to 2 from 2 total entries
    Current filter: <all>

    Group1/
        Name = Group1
        Description =
        MultiplePeersPolicy = RoundRobin/Failover
        PeerTimeOutPolicy = FailOver/SendError/DropPacket
        DiaRemoteServers/
            Entries 1 to 2 from 2 total entries
            Current filter: <all>

            vm023/
                Name = vm023
                Metric = 1
                Weight = 0
                IsActive = TRUE
            vm045/
                Name = vm045
                Metric = 2
                Weight = 0
                IsActive = TRUE
    Group2/
        Name = Group2
        Description =
        MultiplePeersPolicy = Failover/RoundRobin
        PeerTimeOutPolicy = FailOver/SendError/DropPacket
        DiaRemoteServers/
            Entries 1 to 2 from 2 total entries
            Current filter: <all>

            vm052/
                Name = vm052
                Metric = 1
                Weight = 0
                IsActive = TRUE
            vm062/
                Name = vm062
                Metric = 2
                Weight = 0
                IsActive = TRUE

[ //localhost/Radius/Services/diapro ]
    Name = diapro
    Description =
    Type = diameter
    IncomingScript~ =
    OutgoingScript~ =
    MultiplePeersPolicy = GroupFailover(/Failover/RoundRobin/ImsiRangebased)
    ServerGroups/
        Entries 1 to 2 from 2 total entries
        Current filter: <all>

        Group1/
```

```
        Name = Group1
        Metric = 1
        IsActive = TRUE
    Group2/
        Name = Group2
        Metric = 2
        IsActive = TRUE
```

Writing Diameter Application in Prime Access Registrar

Prime Access Registrar supports extensibility by allowing users to create new:

- authentication/authorization applications
- accounting applications
- command codes
- AVP's

This section contains the following topics:

Configuring rex script/service for Diameter

To configure script/service for diameter using aregcmd:

Step 1 Add diameter AVPs in //localhost/Radius/Advanced/DiameterDictionary/DiameterAttributes other than Base stack AVPs.

```
[ //localhost/Radius/Advanced/DiameterDictionary/DiameterAttributes/test ]
    Name = test
    Description =
    Attribute =
    VendorID = 0
    Mandatory = May
    May-Encrypt = No
    Protected = May
    Type =
```

Step 2 Write a rex script (C/C++) and add it in the scripting point or rex service.

```
[ //localhost/Radius/Services/diaservice ]
    Name = diaservice
    Description =
    Type = rex
    IncomingScript~ =
    OutgoingScript~ =
    OutagePolicy~ = RejectAll
    OutageScript~ =
    Filename = librexscript.so
```

```
EntryPoint = DiaService
InitEntryPoint =
InitEntryPointArgs =
```

Refer to Sample rex script/service, page 4-19.

Scripting in Diameter

Prime Access Registrar supports 'rex' scripts for Diameter protocol. The script can be configured only as the server incoming script. The commands available for scripting are restricted to 'get' and 'put' on the dictionaries. While setting a value to an attribute, the following convention needs to be followed "<type number>,<value>". For example, if a 'Class' attribute needs to be added to the response dictionary with value as "classvalue", then set it as follows in the script:
pResponse->put(pResponse, "Class", "1,classvalue", REX_REPLACE);

The following is the list of supported scripting types with the respective type numbers:

```
AVP_STRING_TYPE = 1
AVP_ADDRESS_TYPE = 2
AVP_INTEGER32_TYPE = 3
AVP_UINTEGER32_TYPE = 4
AVP_UTF8_STRING_TYPE = 6
AVP_ENUM_TYPE = 7
AVP_TIME_TYPE = 11
```

Setting response attributes via a script is the only mechanism to add authorization attributes for Diameter requests.

Diameter Environment Variables

This section lists the environment variables that you can use in scripts for Diameter messages.

Table 4-1 lists the Diameter Environment variables and descriptions.

Table 4-1 Diameter Environment Variables

Variable	Description
Request-Type Response-Type	String value. Get/Set the request and response type for diameter packet. **Sample Values** Diameter-Access-Request Diameter-Access-Accept Diameter-Access-Reject Diameter-Accounting-Request Diameter-Accounting-Response Diameter-Proxy-Request Diameter-Proxy-Answer
Diameter-Application-Id	String value. Get the application id for the packet. For setting in response, need to use Auth-Application-id AVPs. **Sample Values** 1 (NASREQ)
Diameter-Command-Code	String value. Get command code for the diameter packet. It will work only for the access-request packet, not for the accounting request. **Sample Values** 265 (AA-Request)

Sample rex script/service

```
int REXAPI DiaService( int iScriptingPoint,
                       rex_AttributeDictionary_t* pRequest,
                       rex_AttributeDictionary_t* pResponse,
                       rex_EnvironmentDictionary_t* pEnviron )
{
    if( iScriptingPoint == REX_START_SERVICE || iScriptingPoint == REX_STOP_SERVICE )
        return REX_OK;
    int iRetVal = REX_ERROR;
    const char* pszRequestType = pEnviron->get( pEnviron, "Request-Type");
    const char* pszAppId = pEnviron->get( pEnviron, "Diameter-Application-Id" );
    const char* pszCmdCode=  pEnviron->get( pEnviron, "Diameter-Command-Code" );
    if(!( pszRequestType && pszAppId && pszCmdCode ))
        return iRetVal;
// check the request type, Application id and command code
/*
Request / Response types
Diameter-Access-Request
Diameter-Access-Accept
Diameter-Access-Reject
Diameter-Accounting-Request
Diameter-Accounting-Response
```

```
*/
    if( (strcmp( pszRequestType, "Diameter-Access-Request") == 0) && (strcmp(
pszAppId,"1") ==0 ) && (strcmp( pszCmdCode,"265\
" )== 0 ) )
    {
// our application
// example how to get DiaAttrib from the packet.
        const char* pszSessionId =  pRequest ->get( pRequest,"Session-Id",0,0 );
// print in trace
        if( pszSessionId )
            pEnviron->trace( pEnviron, 5, "Diameter Session Id: %s", pszSessionId );
// example: how to add dia attrib in response packet
        pResponse->put( pResponse, "Calling-Station-Id", "1,00-01-02-03-05", REX_APPEND );
        pEnviron->put( pEnviron, "Response-Type", "Diameter-Access-Accept");
        iRetVal = REX_OK;
    }
    return iRetVal;
}
```

Traces/Logs

```
05/14/2015  6:11:05.796: P79: Packet received from 10.81.79.81
05/14/2015  6:11:05.796: P79: Application id: 1, Cmd code: 265, Flag: 0x80
05/14/2015  6:11:05.796: P79: Using Client: vm050
05/14/2015  6:11:05.796: P79: Packet successfully added
05/14/2015  6:11:05.796: P79: Trace of Diameter Packet
05/14/2015  6:11:05.796: P79:    Destination-Realm = cisco.com
05/14/2015  6:11:05.796: P79:    User-Name = bob
05/14/2015  6:11:05.796: P79:    User-Password = <encrypted>
05/14/2015  6:11:05.796: P79:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  6:11:05.796: P79:    Origin-Host = ar-lnx-vm050.cisco.com
05/14/2015  6:11:05.796: P79:    Session-Id = .;2096298391;2
05/14/2015  6:11:05.796: P79:    Auth-Application-Id = 1
05/14/2015  6:11:05.796: P79:    Origin-Realm = xyz.com
05/14/2015  6:11:05.796: P79: Tracing the packet after running the rules and policies
05/14/2015  6:11:05.796: P79: Using Client: vm050
05/14/2015  6:11:05.796: P79:  FastRule Engine called for packet
05/14/2015  6:11:05.796: P79: Fastrule return = 0
05/14/2015  6:11:05.796: P79: Authorizing with Service DiaService
05/14/2015  6:11:05.796: P79:     Rex: environ->get( "Request-Type" ) ->
"Diameter-Access-Request"
05/14/2015  6:11:05.797: P79:     Rex: environ->get( "Diameter-Application-Id" ) ->
"1"
05/14/2015  6:11:05.797: P79:     Rex: environ->get( "Diameter-Command-Code" ) ->
"265"
05/14/2015  6:11:05.797: P79:     Rex: request->get( "Session-Id", 0 ) ->
".;2096298391;2"
05/14/2015  6:11:05.797: P79: Diameter Session Id: .;2096298391;2
05/14/2015  6:11:05.797: P79:     Rex: response->put( "Calling-Station-Id",
"1,00-01-02-03-05", 0 ) -> TRUE
05/14/2015  6:11:05.797: P79:     Rex: environ->put( "Response-Type",
"Diameter-Access-Accept" ) -> TRUE
05/14/2015  6:11:05.797: P79: Trace of Diameter Packet
05/14/2015  6:11:05.797: P79:    User-Name = bob
05/14/2015  6:11:05.797: P79:    Result-Code = Diameter-Success
05/14/2015  6:11:05.797: P79:    Auth-Request-Type = AUTHORIZE_ONLY
05/14/2015  6:11:05.797: P79:    Origin-Host = 10.81.79.81
05/14/2015  6:11:05.797: P79:    Session-Id = .;2096298391;2
05/14/2015  6:11:05.797: P79:    Calling-Station-Id = 1,00-01-02-03-05
05/14/2015  6:11:05.797: P79:    Auth-Application-Id = 1
05/14/2015  6:11:05.797: P79:    Origin-Realm = cisco.com
05/14/2015  6:11:05.797: P79: Sending response to ar-lnx-vm050.cisco.com
```

```
05/14/2015  6:11:05.797: P79: Packet successfully removed
05/14/2015  6:11:05.797: P79: Packet Deleted
```

Translation Framework for Diameter

Prime Access Registrar supports translation of an incoming RADIUS request to a Diameter request and vice versa.

The following services are created to set up the translation framework:

- Radius-Diameter—For translation of incoming RADIUS request to Diameter equivalent and then the Diameter response to RADIUS equivalent.
- Diameter-Radius—For translation of incoming Diameter request to RADIUS equivalent and then the RADIUS response to Diameter equivalent.

For both the translation services, Prime Access Registrar uses the following scripting points to operate on the original packet and on the newly translated packet based on request and response mapping:

- PreRequestTranslationScript—To add/modify/delete incoming RADIUS/Diameter attribute values in the request before translation
- PostRequestTranslationScript—To add/modify/delete translated Diameter/RADIUS attributes in the request after translation
- PreResponseTranslationScript—To add/modify/delete Diameter/RADIUS attribute values in the response before translation
- PostResponseTranslationScript—To add/modify/delete RADIUS/Diameter attributes in the response after translation

RADIUS to Diameter translation comes with the 3GPP reverse authorization, if the property is set as True. In that case, the request command mapping must not be defined because the new diameter request is created from the radius request by the 3GPP reverse authorization service. When the diameter response is received from the diameter proxy service, it translates the Diameter response to RADIUS response based on the response mapping configuration and sends radius response to the client.

Prime Access Registrar supports CoA and PoD translation to Re-Auth-Request (RAR) / Abort-Session-Request (ASR), which is triggered directly to Diameter Client without any DRA. Prime Access Registrar sends the translated RAR/ASR packets to client, by configuring a parameter **SendRAR-ASRToClient**. You must also configure the Diameter client to which the packet needs to be sent using the host name of the client in the translation service.

Both these translation services create and maintain appropriate states (with the necessary identifiers, packet pointers, etc) to correlate Request to Response. The states will be cleared if present beyond the ‘Timeout’ property value and all the retries have been exhausted. You can configure the number of retries under Diameter-RemoteServers.

For more information about the translation parameters, see Simple Services, page 2-24.

CLI for RADIUS-Diameter Translation

Following is the CLI for RADIUS to Diameter translation:

```
[ //localhost/Radius/Services/rad-dia-trans ]
    Name = rad-dia-trans
    Description =
    Type = radius-diameter
    SendRAR-ASRToClient = true
    ClientHostName =
```

```
    DiameterApplicationId = 5
    ProxyServiceName = dia
    EnableRequestCommandMappings = true
    PreRequestTranslationScript~ = sm
    PostRequestTranslationScript~ =
    PreResponseTranslationScript~ = env
    PostResponseTranslationScript~ =
    RequestMapping/
        CommandMappings/
            Radius-CoA-Request = Re-Auth
            Radius-PoD-Request = Abort-Session
        AVPMappings/
            Calling-Station-Id = Session-Id
        AVPsToBeAdded/
            Re-Auth-Request-Type = AUTHORIZE_AUTHENTICATE
        EnvironmentMappings/
    ResponseMapping/
        ResultCodeMappings/
            Diameter-Success = Radius-PoD-ACK
            Diameter-Unable-To-Deliver = Radius-PoD-Nak
        AVPMappings/
        AVPsToBeAdded/
        EnvironmentMappings/
```

CLI for Diameter-RADIUS Translation

Following is the CLI for Diameter to RADIUS translation:

```
[ /Radius/Services/dia-rad ]
    Name = dia-rad
    Description =
    Type = diameter-radius
     ProxyServiceName = rad-proxy
    PreRequestTranslationScript~ =
    PostRequestTranslationScript~ = dia-rad-addpassword
    PreResponseTranslationScript~ =
    PostResponseTranslationScript~ = diareadwritetest
    RequestMapping/
        CommandMappings/
            AA = Radius-Access-Request
        AVPMappings/
            Origin-Host = NAS-Identifier
            User-Name = User-Name
        AVPsToBeAdded/
            NAS-Port = 1
        EnvironmentMappings/
    ResponseMapping/
        ResultCodeMappings/
            Radius-Access-Accept = Diameter-Success
            Radius-Access-Reject = Diameter-Unable-To-Deliver
        AVPMappings/
        AVPsToBeAdded/
        EnvironmentMappings/
```

TLS Support for Diameter

Prime Access Registrar supports Transport Level Security (TLS) mechanism for Diameter stack. The system provides an option to enable TLS for Diameter client and Diameter remote server. When the TLS option is disabled, communication is established directly using the transport layer without applying any

encryption. The Diameter TLS feature uses the CiscoSSL libraries, which are available as part of the Prime Access Registrar package.

Following is the CLI configuration of a Diameter client with TLS support:

```
[ /Radius/Clients/vm31 ]
    Name = vm31
    Description =
    Protocol = diameter
    HostName = ar-lnx-vm031.cisco.com
    PeerPort = 3868
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    AdvertisedHostName =
    AdvertisedRealm =
    MaxIncomingRequestRate = 0
    WatchDogTimeout = 500
    SCTP-Enabled = FALSE
    TLS-Enabled = TRUE
    TLSOptions/
        PrivateKeyPassword = cisco
        ServerCertificateFile = /opt/CSCOar/pki/cert.pem
        ServerKeyFile = /opt/CSCOar/pki/key.pem
        CACertificateFile = /opt/CSCOar/pki/root-cert.pem
        CACertificatePath =
        PeerVerificationMode = None/Optional/RequireCertificate
        VerificationDepth = 4
        EnableAutoChaining = True
```

Following is the CLI configuration of a Diameter remote server with TLS support:

```
[ /Radius/RemoteServers/vm58 ]
    Name = vm58
    Description =
    Protocol = diameter
    HostName = ar-lnx-vm058.cisco.com
    Port = 4322
    DestinationRealm = cisco.com
    ReactivateTimerInterval = 300000
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 3
    InitialTimeout = 2000
    LimitOutstandingRequests = FALSE
    MaxPendingPackets = 0
    MaxOutstandingRequests = 0
    DWatchDogTimeout = 2500
    SCTP-Enabled = FALSE
    TLS-Enabled = TRUE
    AdvertiseHostName =
    AdvertiseRealm =
    TLSOptions/
        PrivateKeyPassword = cisco
        ServerCertificateFile = /opt/CSCOar/pki/cert.pem
        ServerKeyFile = /opt/CSCOar/pki/key.pem
        CACertificateFile = /opt/CSCOar/pki/root-cert.pem
        CACertificatePath =
        PeerVerificationMode = None/Optional/RequireCertificate
        VerificationDepth = 4
        EnableAutoChaining = True
```

For descriptions of the TLS options, see the Network Resources, page 2-119 section of Chapter 2, "Using the Graphical User Interface."

Managing Diameter Sessions

Diameter provides two kinds of services namely authentication/authorization and accounting only (optional). Diameter sessions can be created when an authentication/authorization request comes from an access point or when an accounting start comes from an access point. When a Diameter client issues an authentication request, Prime Access Registrar sends the packet with a Session-Id AVP, which can be used to correlate a Diameter message with a user-session. When a Session Termination Request (STR) message is received from the Diameter client, Prime Access Registrar releases the sessions. Also Re-authentication requests must be mapped to the corresponding user session. In case of accounting packets, the session is created when the accounting start is received from the Diameter client. The session is deleted when the accounting stop message is received.

Prime Access Registrar creates a new session when it receives an authentication or accounting request packet from a Diameter client and when a user session is not already present. It allocates the resources for the particular session from the resource manager and stores the session in a session backing store. This session backing store is a file where session information is written. When a session termination message or an accounting stop message comes from the Diameter client, the session data is deleted from the backing store. Apart from this, Prime Access Registrar maintains the session state for every session it creates. Session cache will be supported for grouped AVPs.

For more information on session manager and its support for Diameter client, see SessionManagers, page 2-106.

Blacklisting Support for Diameter Remote Server

Prime Access Registrar supports blacklisting of IMSI or IP address values for Diameter remote servers.

You can choose to configure blacklisting as part of the outgoing script of a Diameter remote server with EAP-SIM or EAP-AKA service. For more information about blacklisting, see .

SCTP Multihoming Support for Diameter Client and Remote Server

Stream Control Transmission Protocol (SCTP) is an IP transport protocol that supports data exchange between exactly two endpoints. Multihoming feature of SCTP provides the ability for a single SCTP endpoint to support multiple IP addresses. With this feature, each of the two endpoints during an SCTP association can specify multiple points of attachment. Each endpoint will be able to receive messages from any of the addresses associated with the other endpoint. With the use of multiple interfaces, data can be sent to alternate addresses when failures occur and thus Prime Access Registrar runs successfully even during network failures.

Prime Access Registrar provides SCTP multihoming support for Diameter client and remote server. With this feature, you can configure multiple source and destination addresses on the Diameter client and remote server.

Note When you use Prime Access Registrar with CentOS, ensure that you configure the Diameter SCTP client and remote servers with different source ports in Prime Access Registrar.

The following shows an example configuration of Diameter remote server with multiple source and destination addresses:

```
[ //localhost/Radius/RemoteServers/Diameter-SCTP-Remote-Server ]
    Name = Diameter-SCTP-Remote-Server
    Description =
    Protocol = diameter
    HostName = 10.197.66.73
    DestinationPort = 3868
    DestinationRealm = cisco.com
    ReactivateTimerInterval = 2000
    Vendor =
    IncomingScript~ =
    OutgoingScript~ =
    MaxTries = 1
    MaxTPSLimit = 0
    MaxSessionLimit = 0
    InitialTimeout = 1500
    LimitOutstandingRequests = FALSE
    MaxPendingPackets = 0
    MaxOutstandingRequests = 0
    DWatchDogTimeout = 2000
    SCTP-Enabled = TRUE
    TLS-Enabled = FALSE
    AdvertiseHostName =
    AdvertiseRealm =
    SCTPParameters/
        SourcePort = 3868
        RTOInitial = 300
        RTOMin = 200
        RTOMax = 300
        MaxInitRetransmits = 8
        AssociationMaxRetrans = 10
        PathMaxRetrans = 10
        RTOCookieLife = 60000
        HBInterval = 50
        SACKTimeout = 400
        InitNumOstreams = 65535
        InitMaxInstreams = 65535
    SCTPAdvertisedHostName/
        Local/
            1. 10.197.66.80
            2. 10.197.66.146
        Remote/
            1. 10.197.66.73
            2. 10.197.66.144
```

The following shows an example configuration of Diameter client with multiple source and destination addresses:

```
[ //localhost/Radius/Clients/Diameter-SCTP-Client ]
   Name = Diameter-SCTP-Client
    Description =
    Protocol = diameter
    HostName = 10.197.66.72
    PeerPort = 3868
    Vendor =
    IncomingScript~ =
```

```
OutgoingScript~ =
AdvertisedHostName =
UserLogEnabled = FALSE
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
KeepAliveTime = 0
SCTP-Enabled = TRUE
TLS-Enabled = FALSE
SCTPParameters/
    SourcePort = 3868
    RTOInitial = 100
    RTOMin = 100
    RTOMax = 100
    MaxInitRetransmits = 8
    AssociationMaxRetrans = 10
    PathMaxRetrans = 5
    RTOCookieLife = 60000
    HBInterval = 50
    SACKTimeout = 200
    InitNumOstreams = 65535
    InitMaxInstreams = 65535
SCTPAdvertisedHostName/
    Local/
        1. 10.197.66.146
        2. 10.197.66.80
    Remote/
        1. 10.197.66.72
        2. 10.197.66.145
```

For details of the SCTP parameters, see SCTPParameters Section, page 2-121.

Diameter Multiple Proxy Support

Prime Access Registrar supports Diameter client configurations in multiple proxy mode. As part of this functionality, client-based Diameter connections can be established from multiple peers with the same IP address but with different source ports and origin-hosts.

The Origin-Host AVP is of type Diameter Identity and must be present in all Diameter messages. This AVP is unique to a host and indicates the endpoint that originated the Diameter message.

When Prime Access Registrar gets a connection from any peer, initially Capabilities Exchange messages (CER-CEA) are exchanged with the client. These messages allow the discovery of peer's identity and its capabilities.

After successful Capabilities exchange with the client, Prime Access Registrar selects the exact client object from the CLI, based on the Origin-Host in CER packet.

A new attribute **EnableMultiProxyMode** is added to the Diameter client configuration to support this feature. To use this feature, you must configure at least two clients in multiple proxy mode, with the same source IP.

Note the following:

- For all the clients configured in multiple proxy mode, the host name must be some name and not an IP address.
- The current implementation of this feature supports only Diameter TCP and TLS connections. It does not support Diameter Routing Agent (DRA) and SCTP connections.

- The maximum number of clients that can be configured in multiple proxy mode with the same IP is 15.
- All the clients configured in multiple proxy mode must have one and the same connection type; either TCP or TLS.

The following CLIs are sample configurations of two clients with same IP Address. **host-1** and **host-2** mentioned in the following samples are host names referring to the same IP address.

```
cli1/
Name = cli1
Description =
Protocol = diameter
EnableMultiProxyMode = TRUE
HostName = host-1
PeerPort = 3868
Vendor =
IncomingScript~ =
OutgoingScript~ =
AdvertisedHostName =
UserLogEnabled =
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
KeepAliveTime = 0
AuthSessionStateInASR = State-Maintained
SCTP-Enabled = FALSE
TLS-Enabled = FALSE

cli2/
Name = cli2
Description =
Protocol = diameter
EnableMultiProxyMode = TRUE
HostName = host-2
PeerPort = 3868
Vendor =
IncomingScript~ =
OutgoingScript~ =
AdvertisedHostName =
UserLogEnabled = FALSE
AdvertisedRealm =
InitialTimeout = 1000
MaxIncomingRequestRate = 0
KeepAliveTime = 0
AuthSessionStateInASR = State-Maintained
SCTP-Enabled = FALSE
TLS-Enabled = FALSE
```

Diameter Overload Indication Conveyance Support for Diameter

Diameter Overload Indication Conveyance (DOIC) is a set of standards for supporting dynamic overload controls between Diameter servers and Diameter clients. This allows Diameter servers to send overload reports to Diameter clients requesting reduction in traffic (throttling) for any duration of time.

Currently Prime Access Registrar has an application-wide throttling mechanism, in which the application starts dropping packets when the incoming request rate is growing above the configured value. Here all the packets are given equal priority. It is not possible to throttle any specific packets.

With the DOIC feature, under active overload conditions, it is possible to throttle (forward, divert, or drop) the packets based on configured priority levels.

There are 2 perspectives when considering DOIC in Prime Access Registrar:

- Prime Access Registrar as reacting node (Client)—when there is a communication between Prime Access Registrar and remote server.
- Prime Access Registrar as reporting node (Server)—When there is a communication between client and Prime Access Registrar.

Prime Access Registrar as Reacting Node

- When Prime Access Registrar pushes traffic to HSS, Prime Access Registrar acts as client and HSS as remote server.
- When both Prime Access Registrar and HSS support DOIC, they exchange capabilities in **OC-Supported-Features** AVP.
- When HSS detects that it is overloaded, it starts sending OC-OLR (Overload Report) AVP in all the response messages; OC-OLR AVP contains the following information about the active overload condition:
  - Overload Reduction Percentage—Indicates the percentage of traffic that the reacting node should throttle.
  - Validity Duration—Indicates the time of expiry of the overload report.
  - Report Type—Indicates the type of overload report, which is **Host_Type**.
  - Sequence Number—Indicates a number, which is incremented every time when any of the sub-AVPs in OC-OLR grouped AVP is modified.
- When Prime Access Registrar receives OC-OLR AVP, it understands that the HSS is overloaded and starts throttling the packets till the specified validity duration. Throttling includes forwarding, diverting, or dropping packets based on configured priority levels.
- Along with throttling, Prime Access Registrar also checks if the outgoing traffic to HSS is reduced by the overload reduction percentage value received in the OC-OLR. After reaching the percentage indicated, further incoming packets will be processed as usual.

Prime Access Registrar as Reporting Node

- When both Prime Access Registrar and the Diameter client support DOIC, they exchange capabilities in **OC-Supported-Features** AVP.
- When Prime Access Registrar reaches 75% of the configured value of **MaxinumNumberOfDiameterPackets**, it becomes overloaded.
- When Prime Access Registrar detects that it is overloaded, it starts sending OC-OLR grouped AVP in all the response messages. This OC-OLR AVP contains information about the active overload condition such as overload reduction percentage, validity duration, report type, and sequence number.

Sample CLI for the DOIC feature

```
A new subfolder DOICPriorities added under /Radius/Advanced

/Radius/Advanced/DOICPriorities

    Priority0/
          Application1/
               ApplicationId =
               CommandCodeList =
```

```
    Priority1/
          Application1/
                ApplicationId =
                CommandCodeList =
    Priority2/
          Application1/
                ApplicationId = 16777245
                CommandCodeList = 303,305
         Application 2/
                ApplicationId =
                CommandCodeList =
    Priority3/
          Application1/
                ApplicationId =
                CommandCodeList =
    Priority4/
          Application1/
                ApplicationId =
                CommandCodeList =
```

The Priority directories are static. The sub directories in it are added dynamically.

```
[ //localhost/Radius/RemoteServers/dia ]
    Name = dia
    Description =
    Protocol = diameter
             :
    DOIC-Enabled = TRUE (can take TRUE/FALSE)
                ForwardAbatement/
          DOICPriorityList = Priority0, Priority1
                DivertAbatement/
          DOICPriorityList =
          HostName =
```

Other than forward and divert, the remaining requests that contribute to overload percentage reduction are dropped.

```
[ //localhost/Radius/Clients/diaclient ]
    Name = diaclient
    Description =
    Protocol = diameter
             :
    DOIC-Enabled =TRUE (can take TRUE/FALSE)
    OverloadReductionPercentage = (value between 0 and 100)
    OLRValidityInSeconds = (value between 1 and 86, 400)
```