ASA 8.0: Configure RADIUS Authentication for WebVPN Users

Contents

Introduction

This document demonstrates how to configure the Cisco Adaptive Security Appliance (ASA) to use a Remote Authentication Dial-In User Service (RADIUS) server for authentication of WebVPN users. The RADIUS server in this example is a Cisco Access Control Server (ACS) server, version 4.1 This configuration is performed with the Adaptive Security Device Manager (ASDM) 6.0(2) on an ASA that runs software version 8.0(2).

Note: In this example RADIUS authentication is configured for WebVPN users, but this configuration can be used for other types of remote access VPN as well. Simply assign the AAA server group to the desired connection profile (tunnel group) as shown.

Prerequisites

- A basic WebVPN configuration is required.
- The Cisco ACS must have users configured for user authentication. Refer to the Adding a Basic User Account section of User Management for more information.

Configure the ACS Server

In this section, you are presented with the information to configure RADIUS authentication on the ACS and ASA.

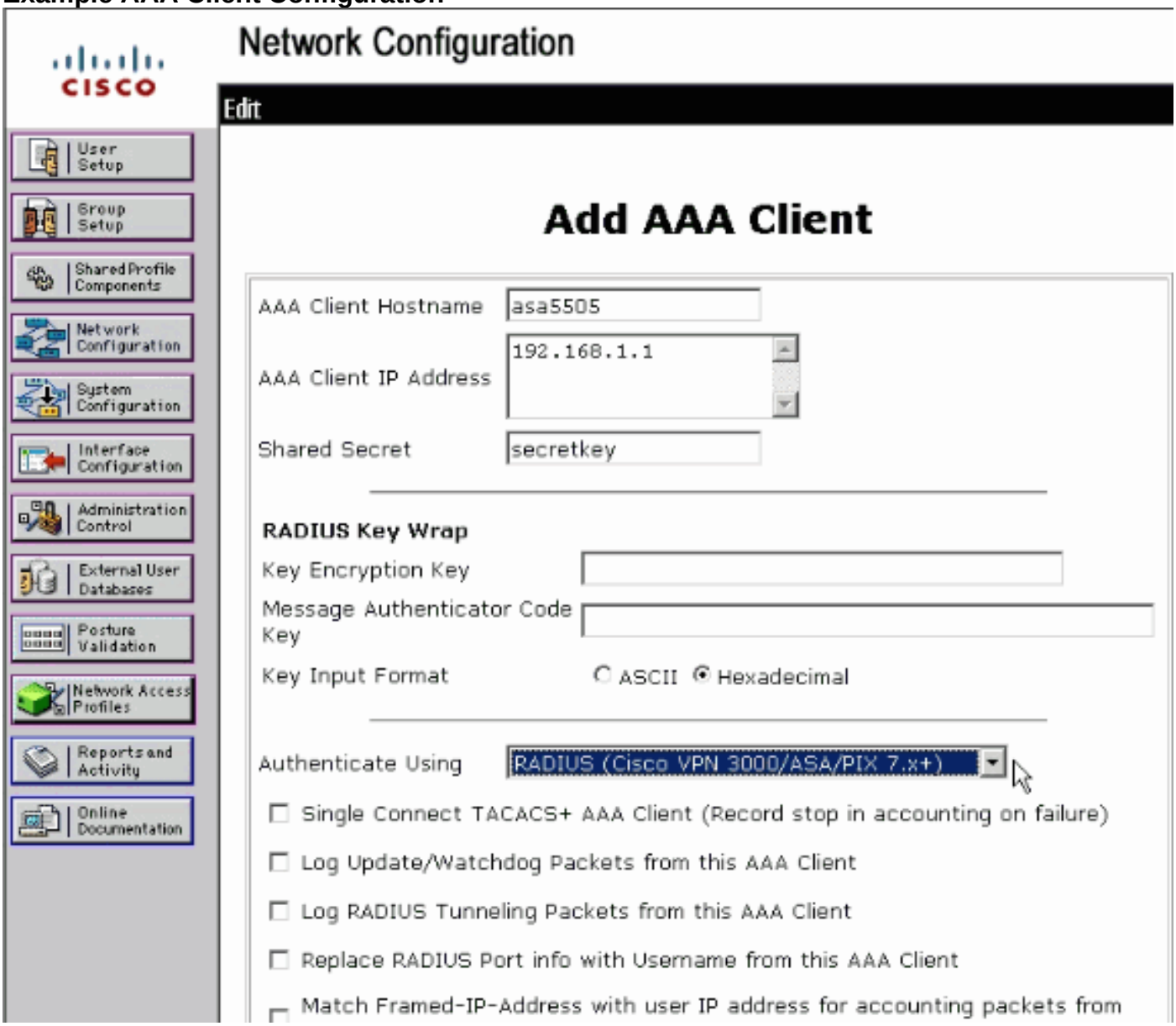
Complete these steps in order to configure the ACS server to communicate with the ASA.

1. Choose **Network Configuration** from the left menu of the ACS screen.
2. Choose **Add Entry** under **AAA Clients**.
3. Provide the client information:**AAA Client Hostname**—a name of your choice**AAA Client IP**

Address—the address from which the security appliance contacts the ACS**Shared Secret**—a secret key configured on the ACS and on the security appliance

4. In the **Authenticate Using** dropdown choose **RADIUS (Cisco VPN 3000/ASA/PIX 7.x+)**.
5. Click **Submit+Apply**.

Example AAA Client Configuration

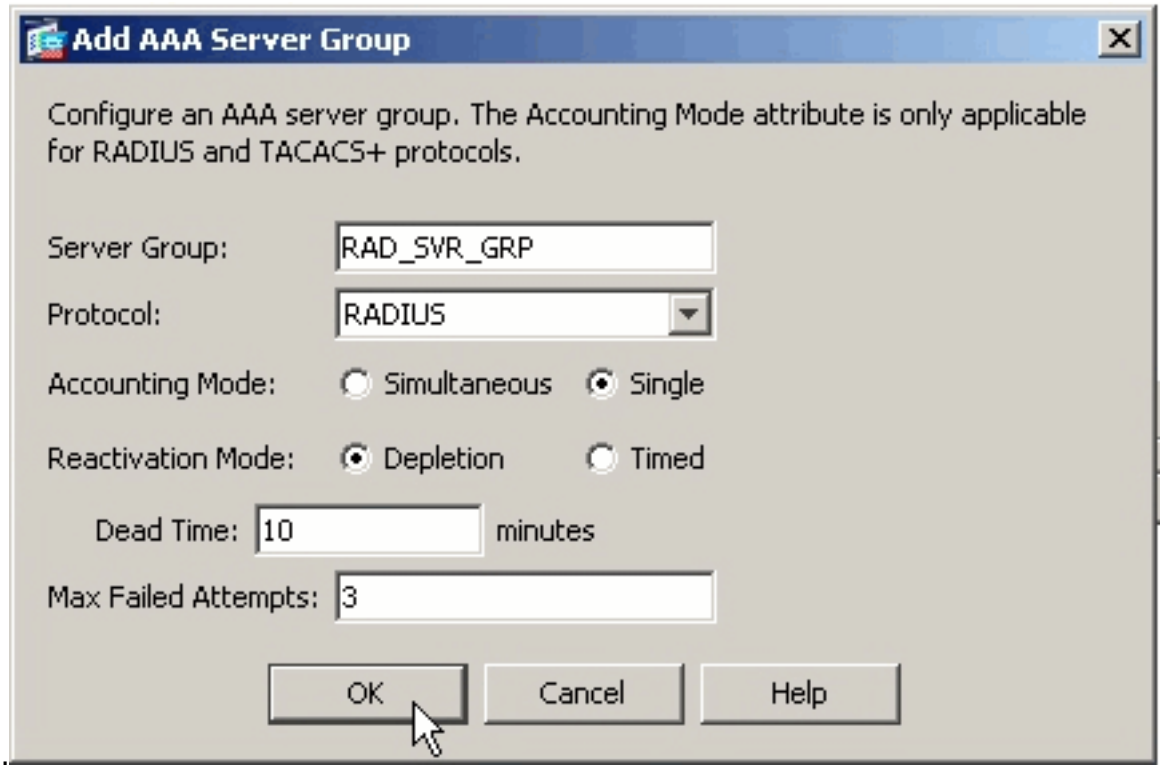


Configure the Security Appliance

ASDM

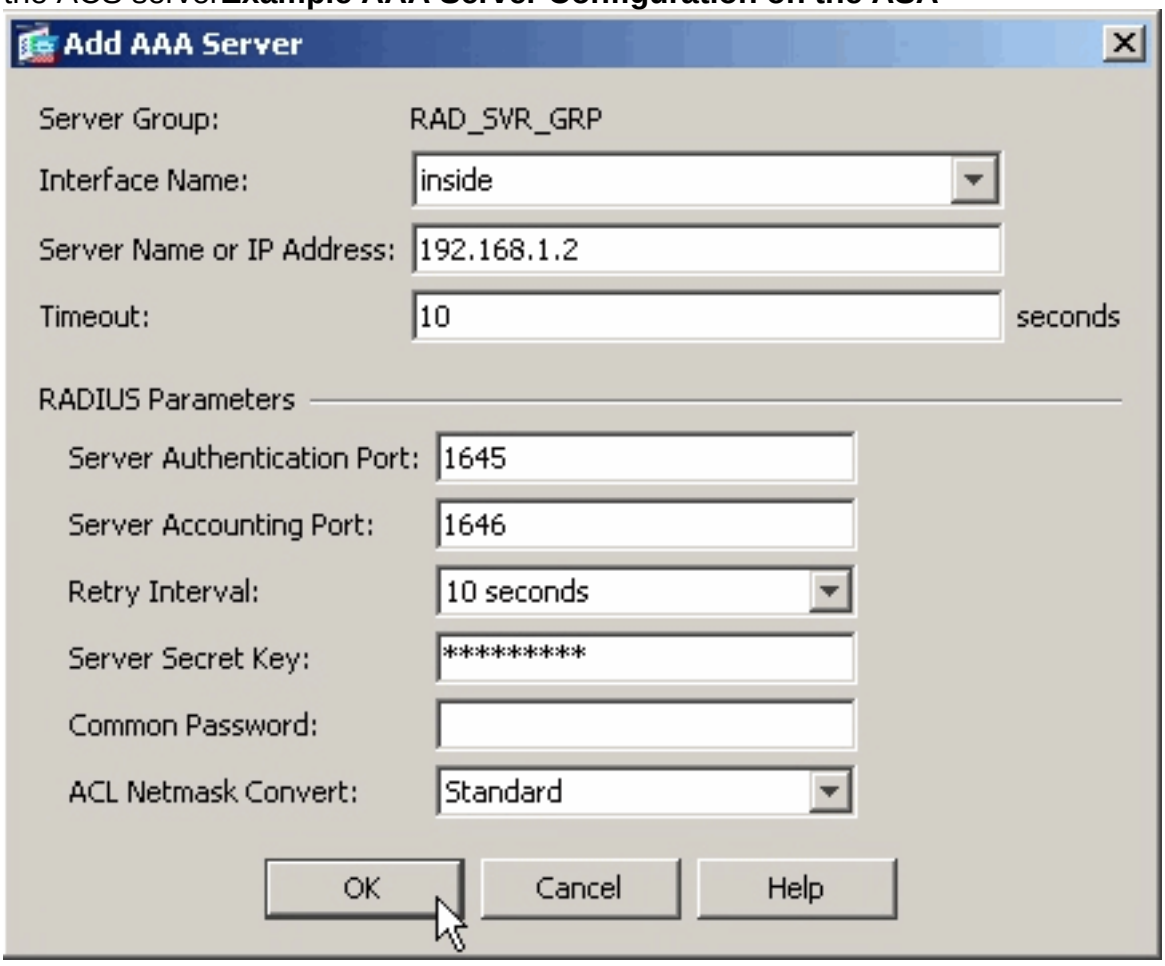
Complete these steps in the ASDM in order to configure the ASA to communicate with the ACS server and authenticate WebVPN clients.

1. Choose **Configuration > Remote Access VPN > AAA Setup > AAA Server Groups**.
2. Click **Add** next to AAA Server Groups.
3. In the window that appears, specify a name for the new AAA Server group and choose **RADIUS** as the protocol. Click **OK** when



finished.

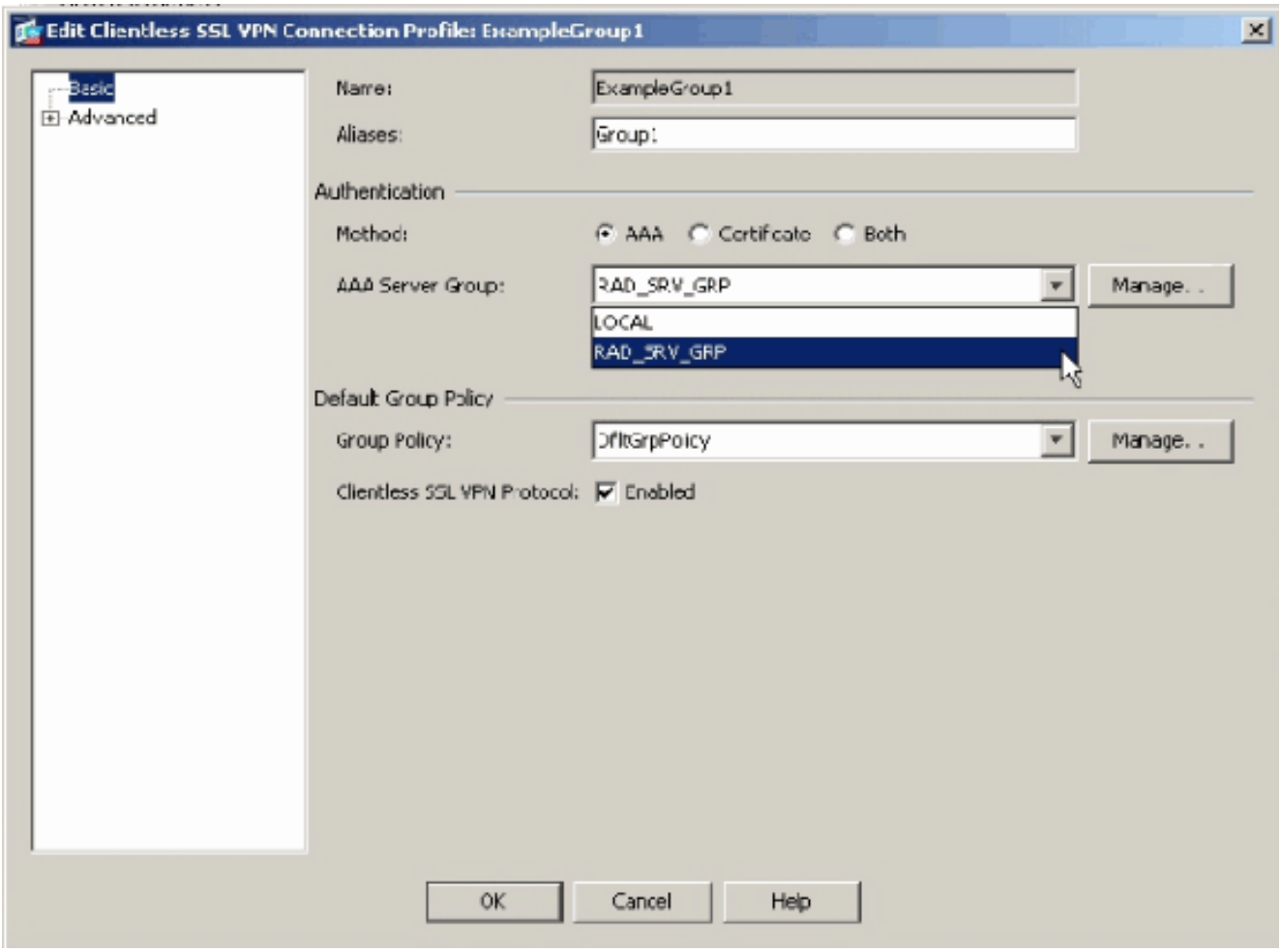
4. Be sure that your new group is selected in the top pane and click **Add** to the right of the lower pane.
5. Provide the server information:**Interface Name**—the interface that the ASA must use to reach the ACS server**Server Name or IP address**—the address that the ASA must use to reach the ACS server**Server Secret Key**—the shared secret key configured for the ASA on the ACS server**Example AAA Server Configuration on the ASA**



6. Once you have configured the AAA server group and server, navigate to Configuration >

Remote Access VPN > Clientless SSL VPN Access > Connection Profiles in order to configure WebVPN to use the new AAA configuration.**Note:** Even though this example uses WebVPN, you can set any remote access connection profile (tunnel group) to use this AAA setup.

7. Choose the profile for which you want to configure AAA, and click **Edit**.
8. Under **Authentication** choose the RADIUS server group that you created earlier. Click **OK** when
finished.



Command Line Interface

Complete these steps in the command line interface (CLI) in order to configure the ASA to communicate with the ACS server and authenticate WebVPN clients.

```
ciscoasa#configure terminal !--- Configure the AAA Server group. ciscoasa(config)# aaa-server
RAD_SRV_GRP protocol RADIUS ciscoasa(config-aaa-server-group)# exit !--- Configure the AAA
Server. ciscoasa(config)# aaa-server RAD_SRV_GRP (inside) host 192.168.1.2 ciscoasa(config-
aaa-server-host)# key secretkey ciscoasa(config-aaa-server-host)# exit !--- Configure the
tunnel group to use the new AAA setup. ciscoasa(config)# tunnel-group ExampleGroup1 general-
attributes ciscoasa(config-tunnel-general)# authentication-server-group RAD_SRV_GRP
```

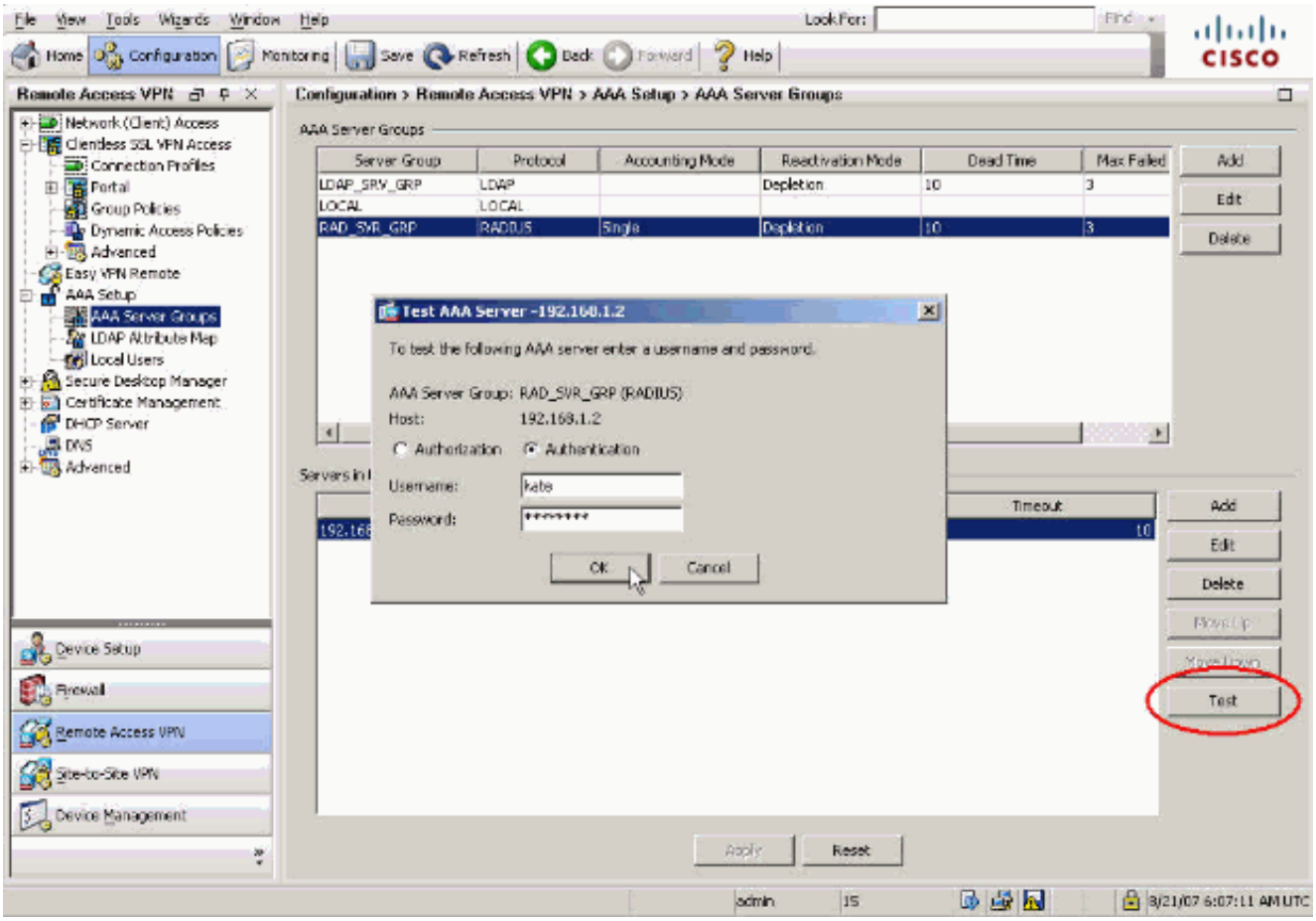
Verify

Use this section in order to confirm that your configuration works properly.

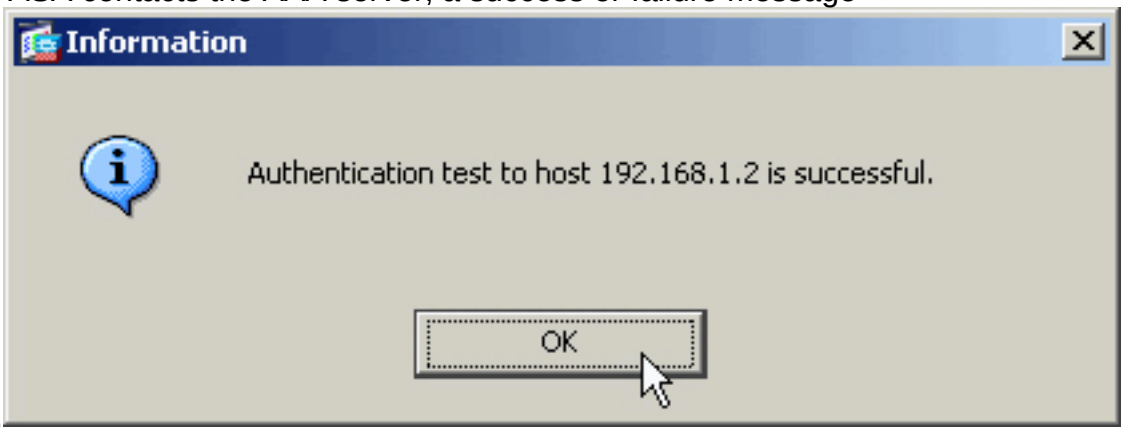
Test with ASDM

Verify your RADIUS configuration with the **Test** button on the AAA Server Groups configuration screen. Once you supply a username and password, this button allows you to send a test authentication request to the ACS server.

1. Choose **Configuration > Remote Access VPN > AAA Setup > AAA Server Groups**.
2. Select your desired AAA Server group in the top pane.
3. Select the AAA server that you want to test in the lower pane.
4. Click the **Test** button to the right of the lower pane.
5. In the window that appears, click the **Authentication** radio button, and supply the credentials with which you want to test. Click **OK** when finished.



6. After the ASA contacts the AAA server, a success or failure message appears.



Test with CLI

You can use the **test** command on the command line in order to test your AAA setup. A test

request is sent to the AAA server, and the result appears on the command line.

```
ciscoasa#test aaa-server authentication RAD_SVR_GRP host 192.168.1.2 username kate password
cisco123 INFO: Attempting Authentication test to IP address <192.168.1.2> (timeout: 12
seconds) INFO: Authentication Successful
```

Troubleshoot

The **debug radius** command can help you to troubleshoot authentication problems in this scenario. This command enables RADIUS session debugging as well as RADIUS packet decoding. In each debug output presented, the first packet decoded is the packet sent from the ASA to the ACS server. The second packet is the response from the ACS server.

Note: Refer to Important Information on Debug Commands before you use **debug** commands.

When authentication is successful, the RADIUS server sends an **access-accept** message.

```
ciscoasa#debug radius !--- First Packet. Authentication Request. ciscoassa#radius mkreq: 0x88
alloc_rip 0xd5627ae4 new request 0x88 --> 52 (0xd5627ae4) got user '' got password add_req
0xd5627ae4 session 0x88 id 52 RADIUS_REQUEST radius.c: rad_mkpkt RADIUS packet decode
(authentication request) ---------------------------------- Raw packet data (length =
62)..... 01 34 00 3e 18 71 56 d7 c4 ad e2 73 30 a9 2e cf | .4.>.qV....s0... 5c 65 3a eb 01 06
6b 61 74 65 02 12 0e c1 28 b7 | \e:...kate....(. 87 26 ed be 7b 2c 7a 06 7c a3 73 19 04 06 c0
a8 | .&..{,z.|.s..... 01 01 05 06 00 00 00 34 3d 06 00 00 00 05 | .......4=..... Parsed
packet data..... Radius: Code = 1 (0x01) Radius: Identifier = 52 (0x34) Radius: Length = 62
(0x003E) Radius: Vector: 187156D7C4ADE27330A92ECF5C653AEB Radius: Type = 1 (0x01) User-Name
Radius: Length = 6 (0x06) Radius: Value (String) = 6b 61 74 65 | kate Radius: Type = 2 (0x02)
User-Password Radius: Length = 18 (0x12) Radius: Value (String) = 0e c1 28 b7 87 26 ed be 7b
2c 7a 06 7c a3 73 19 | ..(..&..{,z.|.s. Radius: Type = 4 (0x04) NAS-IP-Address Radius: Length
= 6 (0x06) Radius: Value (IP Address) = 192.168.1.1 (0xC0A80101) Radius: Type = 5 (0x05) NAS-
Port Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x34 Radius: Type = 61 (0x3D) NAS-Port-
Type Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x5 send pkt 192.168.1.2/1645 rip
0xd5627ae4 state 7 id 52 rad_vrfy() : response message verified rip 0xd544d2e8 : chall_state
'' : state 0x7 : timer 0x0 : reqauth: 18 71 56 d7 c4 ad e2 73 30 a9 2e cf 5c 65 3a eb : info
0x88 session_id 0x88 request_id 0x34 user 'kate' response '***' app 0 reason 0 skey
'secretkey' sip 192.168.1.2 type 1 !--- Second Packet. Authentication Response. RADIUS packet
decode (response) ---------------------------------- Raw packet data (length = 50).....
02 34 00 32 35 a1 88 2f 8a bf 2a 14 c5 31 78 59 | .4.25../..*..1xY 60 31 35 89 08 06 ff ff ff
ff 19 18 43 41 43 53 | `15.........CACS 3a 30 2f 32 61 36 2f 63 30 61 38 30 31 30 31 2f |
:0/2a6/c0a80101/ 35 32 | 52 Parsed packet data..... Radius: Code = 2 (0x02) Radius:
Identifier = 52 (0x34) Radius: Length = 50 (0x0032) Radius: Vector:
35A1882F8ABF2A14C531785960313589 Radius: Type = 8 (0x08) Framed-IP-Address Radius: Length = 6
(0x06) Radius: Value (IP Address) = 255.255.255.255 (0xFFFFFFFF) Radius: Type = 25 (0x19)
Class Radius: Length = 24 (0x18) Radius: Value (String) = 43 41 43 53 3a 30 2f 32 61 36 2f 63
30 61 38 30 | CACS:0/2a6/c0a80 31 30 31 2f 35 32 | 101/52 rad_procpkt: ACCEPT
RADIUS_ACCESS_ACCEPT: normal termination RADIUS_DELETE remove_req 0xd5627ae4 session 0x88 id
52 free_rip 0xd5627ae4 radius: send queue empty
```

When authentication fails, the ACS server sends an **access-reject** message.

```
ciscoasa#debug radius !--- First Packet. Authentication Request. ciscoasa# radius mkreq: 0x85
alloc_rip 0xd5627ae4 new request 0x85 --> 49 (0xd5627ae4) got user '' got password add_req
0xd5627ae4 session 0x85 id 49 RADIUS_REQUEST radius.c: rad_mkpkt RADIUS packet decode
(authentication request) ---------------------------------- Raw packet data (length =
62)..... 01 31 00 3e 88 21 46 07 34 5d d2 a3 a0 59 1e ff | .1.>.!F.4]...Y.. cc 15 2a 1b 01 06
6b 61 74 65 02 12 60 eb 05 32 | ..*...kate..`..2 87 69 78 a3 ce d3 80 d8 4b 0d c3 37 04 06 c0
a8 | .ix.....K..7.... 01 01 05 06 00 00 00 31 3d 06 00 00 00 05 | .......1=..... Parsed
packet data..... Radius: Code = 1 (0x01) Radius: Identifier = 49 (0x31) Radius: Length = 62
(0x003E) Radius: Vector: 88214607345DD2A3A0591EFFCC152A1B Radius: Type = 1 (0x01) User-Name
```

```
Radius: Length = 6 (0x06) Radius: Value (String) = 6b 61 74 65 | kate Radius: Type = 2 (0x02)
User-Password Radius: Length = 18 (0x12) Radius: Value (String) = 60 eb 05 32 87 69 78 a3 ce
d3 80 d8 4b 0d c3 37 | `..2.ix.....K..7 Radius: Type = 4 (0x04) NAS-IP-Address Radius: Length
= 6 (0x06) Radius: Value (IP Address) = 192.168.1.1 (0xC0A80101) Radius: Type = 5 (0x05) NAS-
Port Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x31 Radius: Type = 61 (0x3D) NAS-Port-
Type Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x5 send pkt 192.168.1.2/1645 rip
0xd5627ae4 state 7 id 49 rad_vrfy() : response message verified rip 0xd544d2e8 : chall_state
'' : state 0x7 : timer 0x0 : reqauth: 88 21 46 07 34 5d d2 a3 a0 59 1e ff cc 15 2a 1b : info
0x85 session_id 0x85 request_id 0x31 user 'kate' response '***' app 0 reason 0 skey
'secretkey' sip 192.168.1.2 type 1 !--- Second packet. Authentication Response. RADIUS packet
decode (response) -------------------------------------- Raw packet data (length = 32).....
03 31 00 20 70 98 50 af 39 cc b9 ba df a7 bd ff | .1. p.P.9....... 06 af fb 02 12 0c 52 65 6a
65 63 74 65 64 0a 0d | ......Rejected.. Parsed packet data..... Radius: Code = 3 (0x03)
Radius: Identifier = 49 (0x31) Radius: Length = 32 (0x0020) Radius: Vector:
709850AF39CCB9BADFA7BDFF06AFFB02 Radius: Type = 18 (0x12) Reply-Message Radius: Length = 12
(0x0C) Radius: Value (String) = 52 65 6a 65 63 74 65 64 0a 0d | Rejected.. rad_procpkt:
REJECT RADIUS_DELETE remove_req 0xd5627ae4 session 0x85 id 49 free_rip 0xd5627ae4 radius:
send queue empty
```

Related Information

- **Remote Authentication Dial-In User Service (RADIUS)**
- **Requests for Comments (RFCs)**
- **Technical Support & Documentation - Cisco Systems**