

Configure ACS to Assign a Group Policy at Login using RADIUS

Document ID: 98608

Contents

Introduction

Prerequisites

- Requirements

Background Information

- ASA Group Policy Configuration

Configure ACS to Assign a Group Policy

- Configure ACS

Verify

- Login

- show vpn-sessiondb

Troubleshoot

- Debug the RADIUS Activity

Related Information

Introduction

You can use RADIUS authentication to assign a group policy to a user when a VPN user logs into the Adaptive Security Appliance (ASA). The group policy that is applied is determined by IETF RADIUS Attribute 25 (Class) that is assigned to the user's group in the Access Control Server (ACS).

In this example, users who are in the **Employees** group in the ACS are assigned **ExamplePolicy1** on the ASA, while users who are in the **Contractors** group are assigned **ExamplePolicy2** on the ASA.

The RADIUS server in this example is Cisco Secure ACS version 4.1.

Prerequisites

Requirements

This document requires that a working RADIUS setup is already configured on the ASA. Refer to Document 98594 to learn how to set up a basic RADIUS configuration on the ASA and Cisco Secure ACS.

Background Information

ASA Group Policy Configuration

In order for the ACS and ASA to work together to assign a group policy to a user at login, the ASA must have group policies configured for the RADIUS attribute 25 to correspond to. Group policy configuration is beyond the scope of this document, but this configuration snippet shows what the group policy configuration on the ASA looks like.

```
group-policy ExamplePolicy1 internal
group-policy ExamplePolicy1 attributes
  banner value This is ExamplePolicy1
```

```
group-policy ExamplePolicy2 internal
group-policy ExamplePolicy2 attributes
  banner value This is ExamplePolicy2
```

Currently, the only difference between the two group policies is the banner that is displayed at login. However, many other options can be configured.

Configure ACS to Assign a Group Policy

Configure ACS

Complete these steps in order to configure the IETF RADIUS Attribute 25 (Class) of a group to correspond to a group policy on the ASA.

1. Select **Interface Configuration** from the left menu of the ACS display.
2. Choose **RADIUS (IETF)**.
3. Make sure that **[025] Class** is checked in the Group column.



The screenshot shows a dialog box titled "Interface Configuration". It contains a list of RADIUS attributes, each with a checkbox. The following table represents the visible attributes and their checked status:

Attribute ID	Attribute Name	Checked
[010]	Login-IP-Port	<input type="checkbox"/>
[018]	Reply-Message	<input checked="" type="checkbox"/>
[020]	Callback-Id	<input checked="" type="checkbox"/>
[022]	Framed-Route	<input checked="" type="checkbox"/>
[023]	Framed-IPX-Network	<input checked="" type="checkbox"/>
[024]	State	<input checked="" type="checkbox"/>
[025]	Class	<input checked="" type="checkbox"/>
[027]	Session-Timeout	<input checked="" type="checkbox"/>
[028]	Idle-Timeout	<input checked="" type="checkbox"/>
[029]	Termination-Action	<input checked="" type="checkbox"/>
[033]	Proxy-State	<input checked="" type="checkbox"/>
[034]	Login-LAT-Service	<input checked="" type="checkbox"/>
[035]	Login-LAT-Node	<input checked="" type="checkbox"/>
[036]	Login-LAT-Group	<input checked="" type="checkbox"/>
[037]	Framed-AppleTalk-Link	<input checked="" type="checkbox"/>
[038]	Framed-AppleTalk-Network	<input checked="" type="checkbox"/>
[039]	Framed-AppleTalk-Zone	<input checked="" type="checkbox"/>
[062]	Port-Limit	<input checked="" type="checkbox"/>
[063]	Login-LAT-Port	<input checked="" type="checkbox"/>
[064]	Tunnel-Type	<input checked="" type="checkbox"/>
[065]	Tunnel-Medium-Type	<input checked="" type="checkbox"/>
[066]	Tunnel-Client-Endpoint	<input type="checkbox"/>
[067]	Tunnel-Server-Endpoint	<input type="checkbox"/>

At the bottom of the dialog, there are two buttons: "Submit" and "Cancel".

4. Select **Group Setup** from the left menu.
5. Pick the desired group from the drop-down and click **Edit Settings**.
6. Scroll down to **IETF RADIUS Attributes** and locate **[025] Class**.

7. Check the box next to **[025] Class**.
8. Enter the name of the group policy on the ASA that you want to assign to users of this group when they log in. Use the format "ou=Polycyname;" where you replace "Polycyname" with the name of your group policy. Click **Submit + Restart** when you are done.

Group Setup

Jump To: Access Restrictions

[022] Framed-Route

[023] Framed-IPX-Network

NAS Specifies

Other

[024] State

[025] Class

 ou=ExamplePolicy1;

[027] Session-Timeout

[028] Idle-Timeout

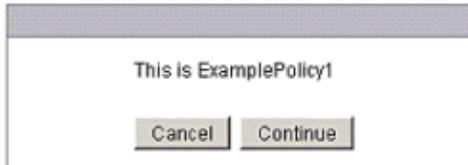
Submit Submit + Restart Cancel

Verify

Use this section to verify your configuration.

Login

One simple way to test if your configuration is successful is to login as a user who is supposed to get a group policy assigned by the ACS. In this example, a banner is configured for each group policy. This screenshot shows a WebVPN user who has logged in successfully and has ExamplePolicy1 applied.



show vpn-sessiondb

The **show vpn-sessiondb** command is extremely useful when you want to verify that your users are assigned the correct group policies. In these examples, **kate** and **ben** use WebVPN, but this **show** command can be used for any kind of remote VPN sessions.

This example output is for **kate** who is in the **Employees** group and is supposed to be assigned **ExamplePolicy1**.

```
ciscoasa#show vpn-sessiondb webvpn

Session Type: WebVPN

Username      : kate                               Index      : 21
Public IP    : 10.88.250.211
Protocol     : Clientless
Encryption   : RC4                               Hashing    : SHA1
Bytes Tx     : 4207283                           Bytes Rx   : 1352862
Group Policy : ExamplePolicy1                 Tunnel Group : ExampleGroup1
Login Time   : 10:24:01 UTC Thu Aug 16 2007
Duration     : 0h:01m:43s
NAC Result   : Unknown
VLAN Mapping : N/A                               VLAN       : none
```

This example output is for **ben** who is in the **Contractors** group and is supposed to be assigned **ExamplePolicy2**.

```
ciscoasa#show vpn-sessiondb webvpn

Session Type: WebVPN

Username      : ben                               Index      : 22
Public IP    : 10.88.250.211
Protocol     : Clientless
Encryption   : RC4                               Hashing    : SHA1
Bytes Tx     : 4331698                           Bytes Rx   : 1373769
Group Policy : ExamplePolicy2                 Tunnel Group : ExampleGroup1
Login Time   : 10:27:25 UTC Thu Aug 16 2007
Duration     : 0h:02m:41s
NAC Result   : Unknown
```

Troubleshoot

Use this section to troubleshoot your configuration.

Debug the RADIUS Activity

When you enable RADIUS debugging you can examine the actual response from the ACS server to make sure that it contains the Class attribute that you desire. The example outputs in this section show that **debug radius** has been enabled. This command enables RADIUS session debugging as well as RADIUS packet decoding. In each debug output presented in this section, 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.

In this output, the ASA contacts the ACS to authenticate the user **kate**. The ACS responds with an **access-accept** as well as the Class attribute that assigns **ExamplePolicy1** to **kate** because she is a member of **Employees** on the ACS server.

```
ciscoasa#debug radius
ciscoasa# radius mkreq: 0x72
alloc_rip 0xd5627ae4
    new request 0x72 --> 36 (0xd5627ae4)
got user ''
got password
add_req 0xd5627ae4 session 0x72 id 36
RADIUS_REQUEST
radius.c: rad_mkpkt

RADIUS packet decode (authentication request)

-----
Raw packet data (length = 113)....
01 24 00 71 df 2c f5 8a fb 18 71 56 d7 c4 ad e2 | .$.q.....qV....
73 30 a9 2e 01 06 6b 61 74 65 02 12 26 1b fb 69 | s0....kate..&..i
2e ec 8d 74 14 b9 8c d8 64 d9 2a 57 1f 0f 31 30 | ...t....d.*W..10
2e 38 38 2e 32 35 30 2e 32 31 31 3d 06 00 00 00 | .88.250.211=....
05 04 06 c0 a8 01 01 05 06 00 00 00 24 1a 24 00 | .....$.$.
00 00 09 01 1e 69 70 3a 73 6f 75 72 63 65 2d 69 | ....ip:source-i
70 3d 31 30 2e 38 38 2e 32 35 30 2e 32 31 31 85 | p=10.88.250.211.
6a | j

Parsed packet data....
Radius: Code = 1 (0x01)
Radius: Identifier = 36 (0x24)
Radius: Length = 113 (0x0071)
Radius: Vector: DF2CF58AFB187156D7C4ADE27330A92E
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) =
26 1b fb 69 2e ec 8d 74 14 b9 8c d8 64 d9 2a 57 | &..i...t....d.*W
Radius: Type = 31 (0x1F) Calling-Station-Id
Radius: Length = 15 (0x0F)
Radius: Value (String) =
31 30 2e 38 38 2e 32 35 30 2e 32 31 31 | 10.88.250.211
Radius: Type = 61 (0x3D) NAS-Port-Type
```

```

Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x5
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) = 0x24
Radius: Type = 26 (0x1A) Vendor-Specific
Radius: Length = 36 (0x24)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 30 (0x1E)
Radius: Value (String) =
69 70 3a 73 6f 75 72 63 65 2d 69 70 3d 31 30 2e      | ip:source-ip=10.
38 38 2e 32 35 30 2e 32 31 31 85 6a                  | 88.250.211.j
send pkt 192.168.1.2/1645
rip 0xd5627ae4 state 7 id 36
rad_vrfy() : response message verified
rip 0xd544d2e8
: chall_state ''
: state 0x7
: timer 0x0
: reqauth:
    df 2c f5 8a fb 18 71 56 d7 c4 ad e2 73 30 a9 2e
: info 0x72
    session_id 0x72
    request_id 0x24
    user 'kate'
    response '***'
    app 0
    reason 0
    skey 'secretkey'
    sip 192.168.1.2
    type 1

```

RADIUS packet decode (response)

```

-----
Raw packet data (length = 70).....
02 24 00 46 cb 46 53 67 3b 5a 77 99 9e c3 91 5e      | $.F.FSg;Zw....^
85 54 70 48 19 14 6f 75 3d 45 78 61 6d 70 6c 65      | .TpH..ou=Example
50 6f 6c 69 63 79 31 3b 08 06 ff ff ff ff 19 18      | Policy1;.....
43 41 43 53 3a 30 2f 31 61 37 2f 63 30 61 38 30      | CACS:0/1a7/c0a80
31 30 31 2f 33 36                                      | 101/36

```

Parsed packet data.....

```

Radius: Code = 2 (0x02)
Radius: Identifier = 36 (0x24)
Radius: Length = 70 (0x0046)
Radius: Vector: CB4653673B5A77999EC3915E85547048
Radius: Type = 25 (0x19) Class
Radius: Length = 20 (0x14)
Radius: Value (String) =
6f 75 3d 45 78 61 6d 70 6c 65 50 6f 6c 69 63 79      | ou=ExamplePolicy
31 3b                                                  | 1;
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 31 61 37 2f 63 30 61 38 30      | CACS:0/1a7/c0a80
31 30 31 2f 33 36                                      | 101/36
rad_procpkt: ACCEPT
RADIUS_ACCESS_ACCEPT: normal termination

```

```

RADIUS_DELETE
remove_req 0xd5627ae4 session 0x72 id 36
free_rip 0xd5627ae4
radius: send queue empty

```

In this output, the ASA contacts the ACS to authenticate the user **ben**. The ACS responds with an **access-accept** as well as the Class attribute that assigns **ExamplePolicy2** to **ben** because he is a member of **Contractors** on the ACS server.

```

ciscoasa#debug radius
ciscoasa# radius mkreq: 0x75
alloc_rip 0xd5627ae4
  new request 0x75 --> 37 (0xd5627ae4)
got user ''
got password
add_req 0xd5627ae4 session 0x75 id 37
RADIUS_REQUEST
radius.c: rad_mkpkt

RADIUS packet decode (authentication request)

-----
Raw packet data (length = 112).....
01 25 00 70 cf 5c 65 3a eb 48 e1 06 c7 f4 1d 92 | .%.p.\e:.H.....
63 60 19 de 01 05 62 65 6e 02 12 a3 6d 71 a2 2c | c`....ben...mq.,
a8 92 ad 5d 19 00 37 d4 c2 8d ca 1f 0f 31 30 2e | ...]..7.....10.
38 38 2e 32 35 30 2e 32 31 31 3d 06 00 00 00 05 | 88.250.211=.....
04 06 c0 a8 01 01 05 06 00 00 00 25 1a 24 00 00 | .....%.$..
00 09 01 1e 69 70 3a 73 6f 75 72 63 65 2d 69 70 | ....ip:source-ip
3d 31 30 2e 38 38 2e 32 35 30 2e 32 31 31 19 45 | =10.88.250.211.E

Parsed packet data.....
Radius: Code = 1 (0x01)
Radius: Identifier = 37 (0x25)
Radius: Length = 112 (0x0070)
Radius: Vector: CF5C653AEB48E106C7F41D92636019DE
Radius: Type = 1 (0x01) User-Name
Radius: Length = 5 (0x05)
Radius: Value (String) =
62 65 6e | ben
Radius: Type = 2 (0x02) User-Password
Radius: Length = 18 (0x12)
Radius: Value (String) =
a3 6d 71 a2 2c a8 92 ad 5d 19 00 37 d4 c2 8d ca | .mq,....]..7....
Radius: Type = 31 (0x1F) Calling-Station-Id
Radius: Length = 15 (0x0F)
Radius: Value (String) =
31 30 2e 38 38 2e 32 35 30 2e 32 31 31 | 10.88.250.211
Radius: Type = 61 (0x3D) NAS-Port-Type
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x5
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) = 0x25
Radius: Type = 26 (0x1A) Vendor-Specific
Radius: Length = 36 (0x24)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 30 (0x1E)
Radius: Value (String) =
69 70 3a 73 6f 75 72 63 65 2d 69 70 3d 31 30 2e | ip:source-ip=10.
38 38 2e 32 35 30 2e 32 31 31 19 45 | 88.250.211.E

```

```

send pkt 192.168.1.2/1645
rip 0xd5627ae4 state 7 id 37
rad_vrfy() : response message verified
rip 0xd544d2e8
 : chall_state ''
 : state 0x7
 : timer 0x0
 : reqauth:
   cf 5c 65 3a eb 48 e1 06 c7 f4 1d 92 63 60 19 de
 : info 0x75
   session_id 0x75
   request_id 0x25
   user 'ben'
   response '***'
   app 0
   reason 0
   skey 'secretkey'
   sip 192.168.1.2
   type 1

```

RADIUS packet decode (response)

```

-----
Raw packet data (length = 70).....
02 25 00 46 2d 78 a3 18 ee fc 2f ac 66 b3 06 33 | .%.F-x..../.f..3
53 31 cf 19 19 14 6f 75 3d 45 78 61 6d 70 6c 65 | S1....ou=Example
50 6f 6c 69 63 79 32 3b 08 06 ff ff ff ff 19 18 | Policy2;.....
43 41 43 53 3a 30 2f 31 61 61 2f 63 30 61 38 30 | CACS:0/1aa/c0a80
31 30 31 2f 33 37 | 101/37

```

```

Parsed packet data.....
Radius: Code = 2 (0x02)
Radius: Identifier = 37 (0x25)
Radius: Length = 70 (0x0046)
Radius: Vector: 2D78A318EEFC2FAC66B306335331CF19
Radius: Type = 25 (0x19) Class
Radius: Length = 20 (0x14)
Radius: Value (String) =
6f 75 3d 45 78 61 6d 70 6c 65 50 6f 6c 69 63 79 | ou=ExamplePolicy
32 3b | 2;
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 31 61 61 2f 63 30 61 38 30 | CACS:0/1aa/c0a80
31 30 31 2f 33 37 | 101/37
rad_procpkt: ACCEPT
RADIUS_ACCESS_ACCEPT: normal termination
RADIUS_DELETE
remove_req 0xd5627ae4 session 0x75 id 37
free_rip 0xd5627ae4
radius: send queue empty

```

Related Information

- [Cisco Secure Access Control Server for Windows](#)
 - [Cisco PIX 500 Series Security Appliances](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

