

Cpmm nested groups

This article will explain how to search within nested groups to determine whether a user is actually a member of a higher level group. This information can then be used to make policy decisions within ClearPass Policy Manager (CPPM).

For example, in my lab, I have three groups in the following configuration:

Group1

Group2

Group3

My user is member of Group3, but I want CPPM to check my membership against Group1. In the real world, you may run into something like:

All Students

ReaganHS

Freshman

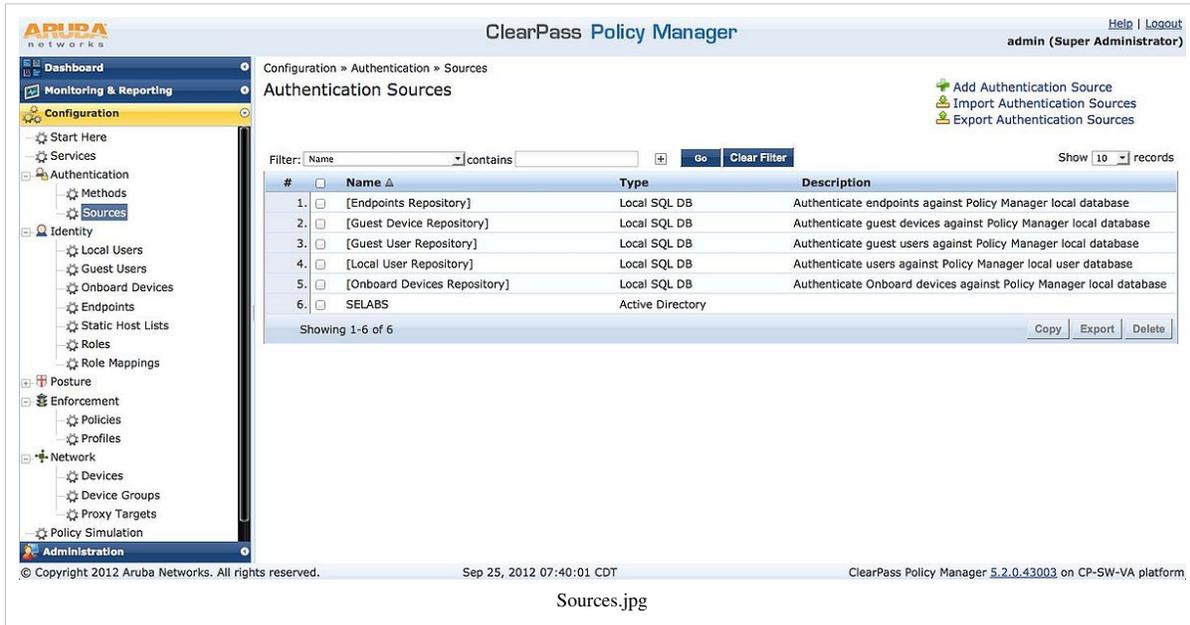
The administrator may want to know if a student is a member of ReaganHS or All Students. Out of the box, CPPM will report that the user is "Freshman", but not the higher level groups.

The default logic of CPPM is to return only the "memberOf" for the user record. Below, we will configure CPPM to use memberOf to again query AD and return the group membership of the group itself. We will then query AD again for the group membership of the second level group. Along the way, we store each set of group memberships in a different variable so that we can use them for policy decisions. In the example above, CPPM will know that Student1 is a member of Freshman and will query AD again to get the group membership of Freshman, which will return ReaganHS. CPPM will then query AD again to get the group membership of ReaganHS, which will return All Students. CPPM will store Freshman, ReaganHS and All Students as different variables.

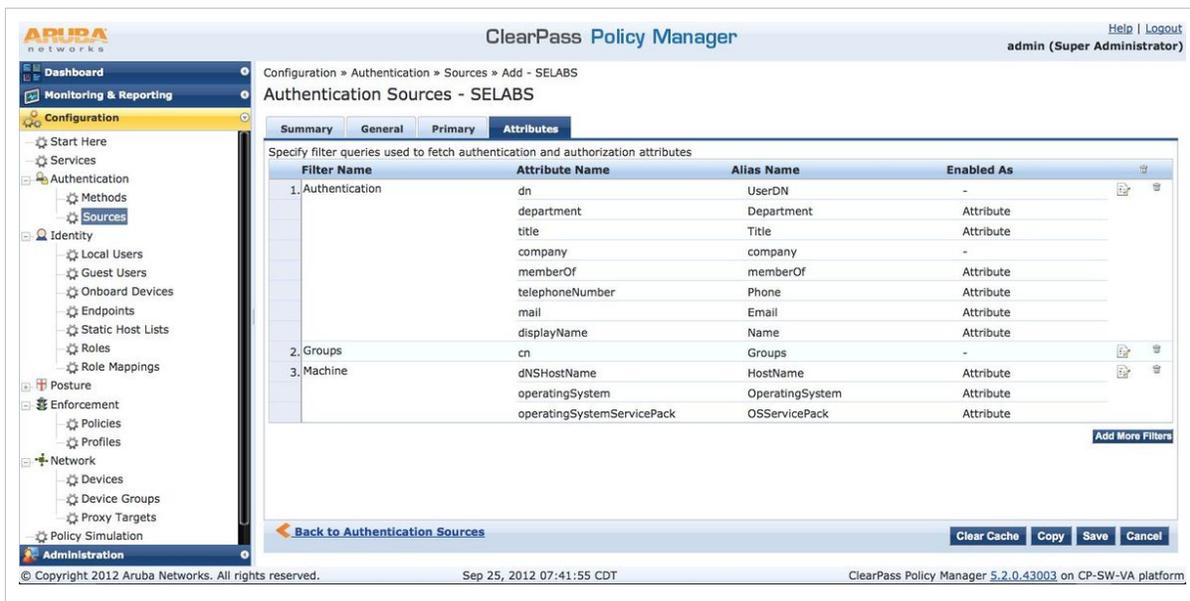
To start, you must already have joined CPPM to your domain and added a domain controller to CPPM as an authentication source. The process for doing these steps is documented in other places and outside the scope of this document.

Once you have successfully joined Active Directory (AD) and added a domain controller as an authentication source, you are ready to look for nested groups.

To start, click on the name of your AD server (SELABS in the example below)



Click on the “Attributes” tab. Your source should already have several filters configured (Authentication, Groups, and Machine) as shown below.



First, edit the filter named “Groups” by clicking on it. Change the Filter Name to “LeafGroups” and the Alias Name to “LeafGroups”. Add another Attribute by clicking on the “Click to add...” link below the existing Attribute. The name must be “memberOf” (it IS case sensitive) and the Alias Name should be “LeafGroupmemberOf”. The screen should now look like the example below.

Configure Filter

Configuration | **Attributes** | **Browse** | **Filter**

Filter Name: LeafGroups

Filter Query: (distinguishedName=%{memberOf})

	Name	Alias Name	Enabled As	
1.	cn	LeafGroups	= -	🗑️
2.	memberOf	LeafGroupmemberOf	= -	🗑️
3.	Click to add...			

Save Close

The Alias Name LeafGroupmemberOf will be referenced later, so remember it.

Make sure you click "Save".

Now, add another filter by clicking the "Add More Filters" button on the bottom right corner of the window. Click the "Configuration" tab on the next window and enter "UpOneLevel" as the Filter Name. In the Filter Query box, enter (without the quotes) "(distinguishedName=%{LeafGroupmemberOf})". This tells the filter to search for the variable called LeafGroupmemberOf, which was set in the initial query of the user record. Click the "Click to add..." link and enter "cn" as the name (again, it is case sensitive) and "UpOneLevel" as the Alias Name. Click the "Click to add..." link again and enter "memberOf" as the name and "UpOneLevelmemberOf" as the Alias Name. The filter should look like the screen shot below.

Configure Filter

Browse | **Filter** | **Attributes** | **Configuration**

Filter Name: UpOneLevel

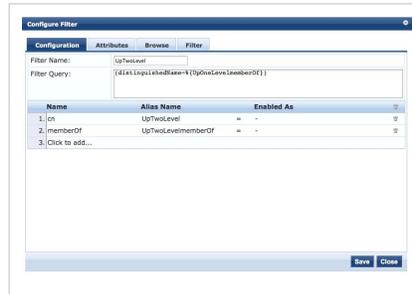
Filter Query: (distinguishedName=%{LeafGroupmemberOf})

	Name	Alias Name	Enabled As	
1.	cn	UpOneLevel	= -	🗑️
2.	memberOf	UpOneLevelmemberOf	= -	🗑️
3.	Click to add...			

Save Close

Click "Save" to save your progress.

Add another filter using the same process above, but make it look like the screen shot below.



Notice that the Filter Query is now looking for “UpOneLevelmemberOf”. The Filter Name and Alias Names have changed as well.

Click “Save”.

If you have more than three levels of nested groups, keep adding levels. For my demo, I only have three.

The final attributes screen should look similar to the one below.

Configuration » Authentication » Sources » Add - SELABS

Authentication Sources - SELABS

Summary General Primary **Attributes**

Filter Name	Attribute Name	Alias Name	Enabled As
1. Authentication	dn	UserDN	-
	department	Department	Attribute
	title	Title	Attribute
	company	company	-
	memberOf	memberOf	Attribute
	telephoneNumber	Phone	Attribute
	mail	Email	Attribute
	displayName	Name	Attribute
2. LeafGroups	cn	LeafGroups	-
	memberOf	LeafGroupmemberOf	-
3. Machine	dNSHostName	HostName	Attribute
	operatingSystem	OperatingSystem	Attribute
	operatingSystemServicePack	OSServicePack	Attribute
4. UpOneLevel	cn	UpOneLevel	-
	memberOf	UpOneLevelmemberOf	-
5. UpTwoLevel	cn	UpTwoLevel	-
	memberOf	UpTwoLevelmemberOf	-

[Back to Authentication Sources](#) Clear Cache Copy Save Cancel

Sources-after edits.jpg

You must configure your authentication source for authorization before CPPM will report authorization attributes. This is done by clicking on “Authentication”, “Sources”, then the “General” tab. Make sure you have checked the “Use for Authorization:” box as shown below.

Configuration » Authentication » Sources » Add - SELABS

Authentication Sources - SELABS

Summary General Primary Attributes

Name: SELABS

Description:

Type: Active Directory

Use for Authorization: Enable to use this authentication source to also fetch role mapping attributes

Authorization Sources:

Server Timeout: 10 seconds

Cache Timeout: 36000 seconds

Backup Servers Priority:

Back to Authentication Sources Clear Cache Copy Save Cancel

Sources with authorization.jpg

CPPM also requires that you add Role Mapping to your Service before the attributes will be recorded.

To add Role Mapping, click on Services and edit the Service that uses the Authentication Source you modified above. In my lab, it is called "oolson-dot1 ssid".

Click the Service tab and ensure Authorization is enabled in the More Options box.

ARUBA networks ClearPass Policy Manager Help | Logout admin (Super Administrator)

Configuration » Services » Edit - oolson-dot1x ssid

Services - oolson-dot1x ssid

Summary Service Authentication Authorization Roles Posture Enforcement

Name: oolson-dot1x ssid

Description: Aruba 802.1X Wireless Access Service

Type: Aruba 802.1X Wireless

Status: Enabled

Monitor Mode: Enable to monitor network access without enforcement

More Options: Authorization Posture Compliance Audit End-hosts Profile Endpoints

Service Rule

Matches ANY or ALL of the following conditions:

Type	Name	Operator	Value
1. Radius:IETF	NAS-Port-Type	EQUALS	Wireless-802.11 (19)
2. Radius:IETF	Service-Type	BELONGS_TO	Login-User (1), Framed-User (2), Authenticate-Only (8)
3. Radius:Aruba	Aruba-Location-Id	EXISTS	
4. Radius:Aruba	Aruba-Essid-Name	EQUALS	oolson-dot1x
5. Click to add...			

Back to Services Disable Copy Save Cancel

© Copyright 2012 Aruba Networks. All rights reserved. Sep 25, 2012 07:54:40 CDT ClearPass Policy Manager 5.2.0.43003 on CP-SW-VA platform

Services-make sure auth enabled.jpg

Click the Authentication tab and ensure your modified AD server is listed first in the Authentication Sources box.

The screenshot shows the ClearPass Policy Manager interface. The left sidebar contains navigation options: Dashboard, Monitoring & Reporting, Configuration, and Administration. Under Configuration, there are sub-menus for Start Here, Services, Authentication (Methods, Sources), Identity (Local Users, Guest Users, Onboard Devices, Endpoints, Static Host Lists, Roles, Role Mappings), Posture, Enforcement (Policies, Profiles), and Network (Devices, Device Groups, Proxy Targets). The main content area is titled 'Configuration » Services » Edit - oolson-dot1x ssid' and 'Services - oolson-dot1x ssid'. It features several tabs: Summary, Service, Authentication, Authorization, Roles, Posture, and Enforcement. The Authentication tab is active, showing a list of Authentication Methods (EAP PEAP, EAP MSCHAPv2, MSCHAP) and Authentication Sources (SELABS (Active Directory), Local User Repository (Local SQL DB)). Each item has a vertical toolbar with actions like Move Up, Move Down, Remove, View Details, and Modify. There are also links for 'Add new Authentication Method' and 'Add new Authentication Source'. At the bottom, there is a 'Strip Username Rules' section with a checkbox to 'Enable to specify a comma-separated list of rules to strip username prefixes or suffixes'. The footer includes a 'Back to Services' link and buttons for Disable, Copy, Save, and Cancel. The footer text reads: '© Copyright 2012 Aruba Networks. All rights reserved. Sep 25, 2012 07:55:59 CDT ClearPass Policy Manager 5.2.0.43003 on CP-SW-VA platform'.

Services-make sure authentication is correct.jpg

To add a Role Mapping, click the Roles tab. Click the “Add new Role Mapping Policy” link.

The screenshot shows the ClearPass Policy Manager interface with the Roles tab selected. The main content area is titled 'Configuration » Services » Edit - oolson-dot1x ssid' and 'Services - oolson-dot1x ssid'. The Roles tab is active, showing a 'Role Mapping Policy' dropdown menu with a '--Select--' option and a 'Modify' button. There is a link for 'Add new Role Mapping Policy'. Below this is a 'Role Mapping Policy Details' section with a table for Description, Default Role, and Rules Evaluation Algorithm, all with '-' values. At the bottom, there is a table with columns 'Conditions' and 'Role'. The footer includes a 'Back to Services' link and buttons for Disable, Copy, Save, and Cancel. The footer text reads: '© Copyright 2012 Aruba Networks. All rights reserved. Sep 25, 2012 07:55:59 CDT ClearPass Policy Manager 5.2.0.43003 on CP-SW-VA platform'.

Services-add new role mapping.jpg

Add a Role Mapping Policy called “Check_Nested_Groups”. I added a new TIPs role called “Group1_Member”, but you can add what ever TIPs roles that make sense to you. The role checks “Authorization:SELABS:UpTwoLevelmemberOf” for “Group1” and assigns Group1_Member TIPs role if it matches.

Configuration » Services » Edit - oolson-dot1x ssid

Services - oolson-dot1x ssid

Summary Service Authentication Authorization **Roles** Posture Enforcement

Role Mapping Policy: [Add new Role Mapping Policy](#)

Role Mapping Policy Details

Description:

Default Role: Group1_Member

Rules Evaluation Algorithm: first-applicable

Conditions	Role
1. (Authorization:SELABS:UpTwoLevelmemberOf CONTAINS Group1)	Group1_Member

[Back to Services](#)

Services-after add new role mapping.jpg

To verify its working, create an AD account and several groups. Make Group3 a member of Group2 and Group2 a member of Group1. Then, make your AD account a member of Group3.

Once you have the configuration above (both CPPM and AD) and you can successfully authenticate against AD via CPPM, you will be able to see the following logs in the CPPM Access Tracker.

Request Details

Summary	Input	Output
Session Identifier:	A0000326-06-5061aaf	
Date and Time:	Sep 25, 2012 08:00:47 CDT	
End-Host Identifier:	CBCC084361EE	
Username:	oolson	
Access Device IP/Port:	192.168.1.1:0	
System Posture Status:	UNKNOWN (100)	
Policies Used		
Service:	oolson-dot1x ssid	
Authentication Method:	EAP-PEAP-EAP-MSCHAPv2	
Authentication Source:	AD:192.168.1.50	
Authorization Source:	SELABS	
Roles:	Group1_Member, (User Authenticated)	
Enforcement Profile:	[Allow Access Profile]	
Service Monitor Mode:	Disabled	

AccessTracker-summary.jpg

Notice that the Roles: in the output above include "Group1_Member". Group1_Member was assigned since my account is a member of Group3, which is a member of Group2, which is a member of Group1.

Request Details

Summary	Input	Output
Access Device IP/Port:	192.168.1.1:0	
RADIUS Request		
Authorization Attributes		
Authorization:SELABS-LeafGroupmemberOf	CN=Group2,CN=Users,DC=selsabs,DC=com	
Authorization:SELABS-LeafGroups	Administrators, Group3, Remote Desktop Users	
Authorization:SELABS-memberOf	CN=Administrators,CN=Builtin,DC=selsabs,DC=com, CN=Group3,CN=Users,DC=selsabs,DC=com, CN=Remote Desktop Users,CN=Builtin,DC=selsabs,DC=com	
Authorization:SELABS-Name	oolson	
Authorization:SELABS-upOneLevel	Group2	
Authorization:SELABS-upOneLevelmemberOf	CN=Group1,CN=Users,DC=selsabs,DC=com	
Authorization:SELABS-upTwoLevel	Group1	
Authorization:SELABS-userDN	CN=oolson,DC=Users,DC=selsabs,DC=com	
Computed Attributes		

AccessTracker-input.jpg

In the output above, you can see that there are several Authorization attributes, including Authorization:SELABS:UpTwoLevel = "Group1".