RADIUS Groups

Contents

Overview

RADIUS Groups.

Swivel can include group information within the RADIUS response for a successful authentication. This can be used by the VPN to allocate different access rights to different user groups.

Different VPNs require this information in different format. Swivel supports a number of formats and new formats can be added if required, with some development work.

Note that the response will reflect the membership of the groups defined on Swivel rather than groups names within the repository

Prerequisites

Swivel 3.6 onwards

RADIUS Group Configuration

On the Swivel RADIUS->NAS screen set if present, set RADIUS groups to Yes.

If you set a RADIUS keyword, only group names that contain that keyword will be returned in the RADIUS response. If this is left blank then all groups may be passed back, depending on Vendor setting.

If a RADIUS authentication is successful then the RADIUS response will include the groups that the user is a member of in the format specified by the Vendor setting.

Group Configuration

Swivel groups can be defined on the Swivel Administration console under Repository/Groups. A group may consist of users from more than one repository data source. Multiple groups can be created with users from one or more repository data sources.

Vendor Settings

Cisco

The list of groups the user is a member of is returned in the RADIUS CLASS attribute (Attribute 25). It is returned in the format "OU=group1;group2"

Fortinet

Only one group is passed back, so use of the Group keyword is required to ensure the correct group is returned. The group is passed back using a Vendor Specific Attribute List (12356).

The attributes in this list are

1. the group
2. the source IP address of the request
3. "root"

Watchguard

Only one group is passed back, so use of the Group keyword is required to ensure the correct group is returned. The group the user is a member of is returned in the RADIUS Filter_ID attribute (Attribute 11). It is returned in the format "OU=group1;group2"

Palo Alto

Adds a single group as attribute ID 5

Single Group

Adds a single group using the standard class attribute (ID=25)

Cisco ACL

Uses a defined attribute to determine the Access Control List (ACL) to return. See 3.11.4 Release Notes for further information.

Sonicwall

Adds the groups as a comma-separated list, to the Filter Id attribute (ID=11)

New Vendor Classes

Support for additional vendors can be achieved by creating new "vendor classes". This requires a knowledge of the Java programming language. You are advised to contact support@swivelsecure.com for further advice before starting to implement such a class.

Create the Class

To create such a custom class, create a class in the com.swiveltechnologies.pinsafe.server.radius.vendor package that extends the AbstractVendor class and implements the Vendor interface.

For example

```
package com.swiveltechnologies.pinsafe.server.radius.vendor;

import java.util.List;

import com.swiveltechnologies.pinsafe.server.user.PINsafeUser;
import com.theorem.radserver3.Attribute;
import com.theorem.radserver3.AttributeList;
import com.theorem.radserver3.AuthInfo;

public class ExampleVendor extends AbstractVendor implements Vendor {

    public Cisco() {
        super();
    }
```

Refer to http://www.axlradius.com/ For information on the RADIUS classes used

The interface has two methods

```
    public void setAclManager(AclManager aclm);
    public AttributeList getVendorAttributeList(PINsafeUser user, String filter, AuthInfo ai)
```

The first method is implemented by AbstractVendor, and there is typically no need to re-implement it. It is currently only used by the Cisco ACL vendor class, but other vendor classes could make use of it. The Vendor Class must implement the second method.

The AbstractVendor class has two additional methods in it:

```
   protected List<String> getFilteredGroupList(PINsafeUser user, String filter);
   public AclManager getAccessControlManager();
```

The first method that returns all the groups the user is a member of, subject to the configured prefixes. The second method returns the Access Control Manager set previously. The latter is currently only used by the Cisco ACL class.

For example

```
public AttributeList getVendorAttributeList(PINsafeUser user, String filter, AuthInfo ai) {
       String param = "";
       AttributeList aList = new AttributeList();

       List<String> group = getFilteredGroupList(user, filter);
       if (group.size() > 0) {
           for (int c = 0; c < group.size(); c++) {
               param = param.concat(group.get(c)) + ";";
           }
           aList.addAttribute(Attribute.Class, param);
           return aList;
       } else {
           return null;
       }
   }
```

Upload the Class

Upload the new class using a program such as WinSCP to pinsafe/WEB-INF/classes/com/swiveltechnologies/pinsafe/server/radius/vendor

Appliance: /usr/local/tomcat/webapps/pinsafe/WEB-INF/classes/com/swiveltechnologies/pinsafe/server/radius/vendor

Register the class

Once the new class has been created it needs to be registered on Swivel.

To do this copy the class to the appropriate path then edit the vendor.properties file under pinsafe/WEB-INF/classes and add the name=class pair as required.

Appliance: /usr/local/tomcat/webapps/pinsafe/WEB-INF/classes/vendor.properties

```
null=com.swiveltechnologies.pinsafe.server.radius.vendor.Null
cisco=com.swiveltechnologies.pinsafe.server.radius.vendor.Cisco
fortinet=com.swiveltechnologies.pinsafe.server.radius.vendor.Fortinet
watchguard=com.swiveltechnologies.pinsafe.server.radius.vendor.Watchguard
paloalto=com.swiveltechnologies.pinsafe.server.radius.vendor.Paloalto
singlegroup=com.swiveltechnologies.pinsafe.server.radius.vendor.SingleGroup
sonicwall=com.swiveltechnologies.pinsafe.server.radius.vendor.Sonicwall
ciscoacl=com.swiveltechnologies.pinsafe.server.radius.vendor.CiscoAcl
example=com.swiveltechnologies.pinsafe.server.radius.vendor.ExampleVendor
```

Restart tomcat for the changes to take effect.