Amigopod

External Authentication Servers

Software Walkthrough

Technical Note

Copyright

Open Source Code

Certain Aruba products include Open Source software code developed by third parties, including software code subject to the GNU General Public License (GPL), GNU Lesser General Public License (LGPL), or other Open Source Licenses. The Open Source code used can be found at this site:

http://www.arubanetworks.com/open_source

Legal Notice

The use of Aruba Networks, Inc. switching platforms and software, by all individuals or corporations, to terminate other vendors' VPN client devices constitutes complete acceptance of liability by that individual or corporation for this action and indemnifies, in full, Aruba Networks, Inc. from any and all legal actions that might be taken against it with respect to infringement of copyright on behalf of those vendors.

Warranty

This hardware product is protected by the standard Aruba warranty of one year parts/labor. For more information, refer to the ARUBACARE SERVICE AND SUPPORT TERMS AND CONDITIONS.

Altering this device (such as painting it) voids the warranty.

ARUBA®
networks

www.arubanetworks.com
1344 Crossman Avenue
Sunnyvale, California 94089
Phone: 408.227.4500
Fax 408.227.4550

Table of Contents

1 External Authentication Servers

About RADIUS Authentication Servers

Authentication is the verification of a user's credentials, typically a username and password.

Many networks have more than one place where user credentials are stored. Networks that have different types of user, geographically separate systems, or networks created by integrating different types of systems are all situations where user account information can be spread across several places.

However, network access equipment is often shared between all of these users. This requires that different authentication sources be integrated for use by the network infrastructure.

The Amigopod RADIUS server supports multiple external authentication servers, allowing user accounts from different places to be authenticated using a common industry-standard interface (RADIUS requests).

Types of authentication server

An authentication server may be one of four types:

- **Local** user database — User accounts defined in Amigopod Guest Manager
- Microsoft Active Directory — User accounts defined in a forest or domain and authenticated by the domain controller
- LDAP server (Lightweight Directory Access Protocol) — User accounts stored in a directory
- **Proxy RADIUS server** — User accounts authenticated by another RADIUS server

Authorization for external authentication servers

Authorization controls the type of access that an authenticated user is permitted to have.

In the context of a RADIUS request being processed by the server, there are two aspects to user authorization:

- **Is the user allowed?** Yes/no decisions can be made in the context of authorization. Examples: user account not enabled; user account expired; user account exceeded a traffic quota within a certain time window.
- What are the user's permitted limits? These are not yes/no decisions, but might involve a calculation based on previous usage (e.g. via the accounting-based authorization functions), or based on properties of a user account (e.g. maximum session lifetime is based on the expiration time for the account)

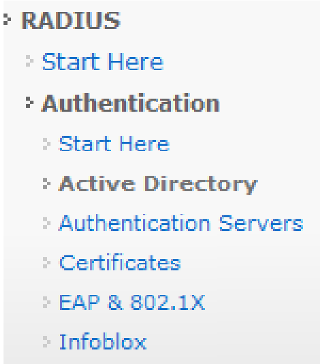
Each type of authentication server has different methods for determining user authorization:

- **No authorization** — Authenticate only may be used to provide a basic user authentication service. The RADIUS server will respond with an Access-Accept or Access-Reject for the authentication attempt. Only RADIUS attributes directly related to user authentication will be returned; all other attributes will be ignored.

- **Use role** assigned to local user is the only authorization method available for the local user database. If the user's authentication attempt is successful, the RADIUS server will respond with an Access-Accept message that includes the RADIUS attributes defined for the user's role.
- Use attributes from Proxy RADIUS server is an authorization method available only for Proxy RADIUS servers. The RADIUS attributes returned by the external RADIUS server are returned unmodified.
- Assign a fixed user role may be used to assign all authenticated users to a particular user role. If the user's authentication attempt is successful, the RADIUS server will respond with an Access-Accept message that includes the RADIUS attributes defined for the fixed role that has been selected for this authentication server.

Configuring RADIUS Authentication

The **RADIUS** > **Authentication** menu contains links to the screens related to configuring authentication:



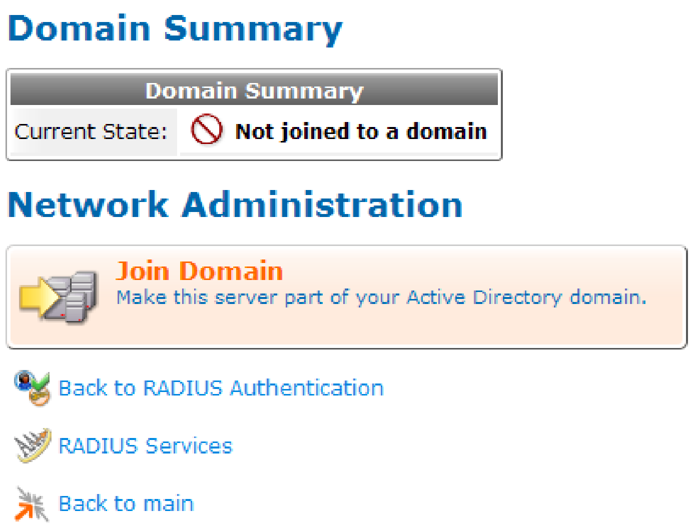
NOTE The RADIUS server's EAP & 802.1X functionality is now located under the **RADIUS** > **Authentication** menu. In earlier software releases, this menu option was located directly below the RADIUS Services heading.

Joining an Active Directory domain

To perform certain types of user authentication, such as using the MS-CHAPv2 protocol to verify a username and password, the RADIUS server must join the domain.

The steps required to join the domain are shown below:

Navigate to **RADIUS** > **Authentication** > **Active Directory**. The following screen will be displayed:



Domain Summary

Domain Summary	
Current State:	⃠ Not joined to a domain

Network Administration

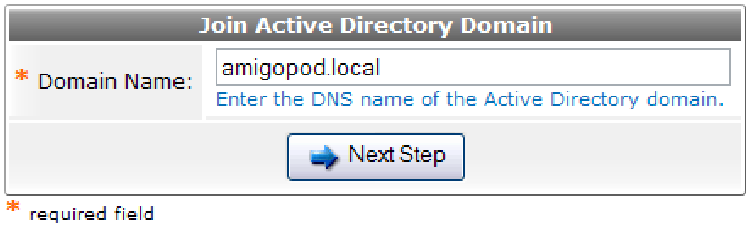
Join Domain
Make this server part of your Active Directory domain.

Back to RADIUS Authentication

RADIUS Services

Back to main

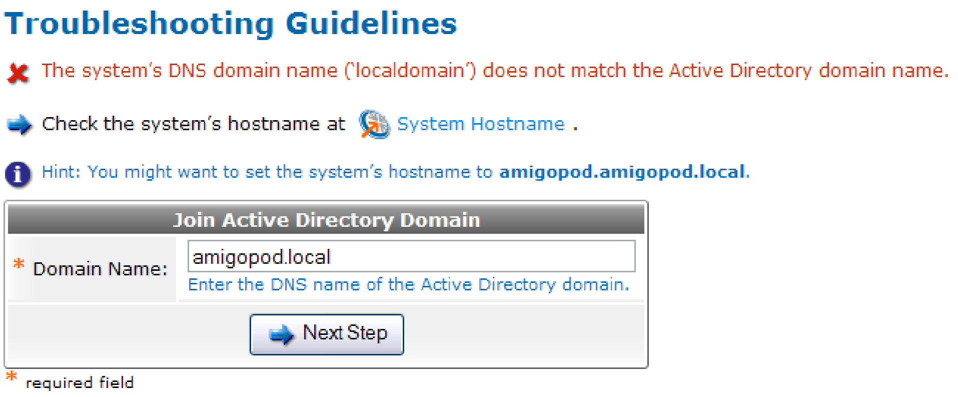
Selecting the Join Domain command starts a two-step process to join the domain:



Join Active Directory Domain	
* Domain Name:	amigopod.local Enter the DNS name of the Active Directory domain.

* required field

The process has built-in troubleshooting assistance, which can help with much of the necessary configuration:



Troubleshooting Guidelines

✗ The system's DNS domain name ('localdomain') does not match the Active Directory domain name.

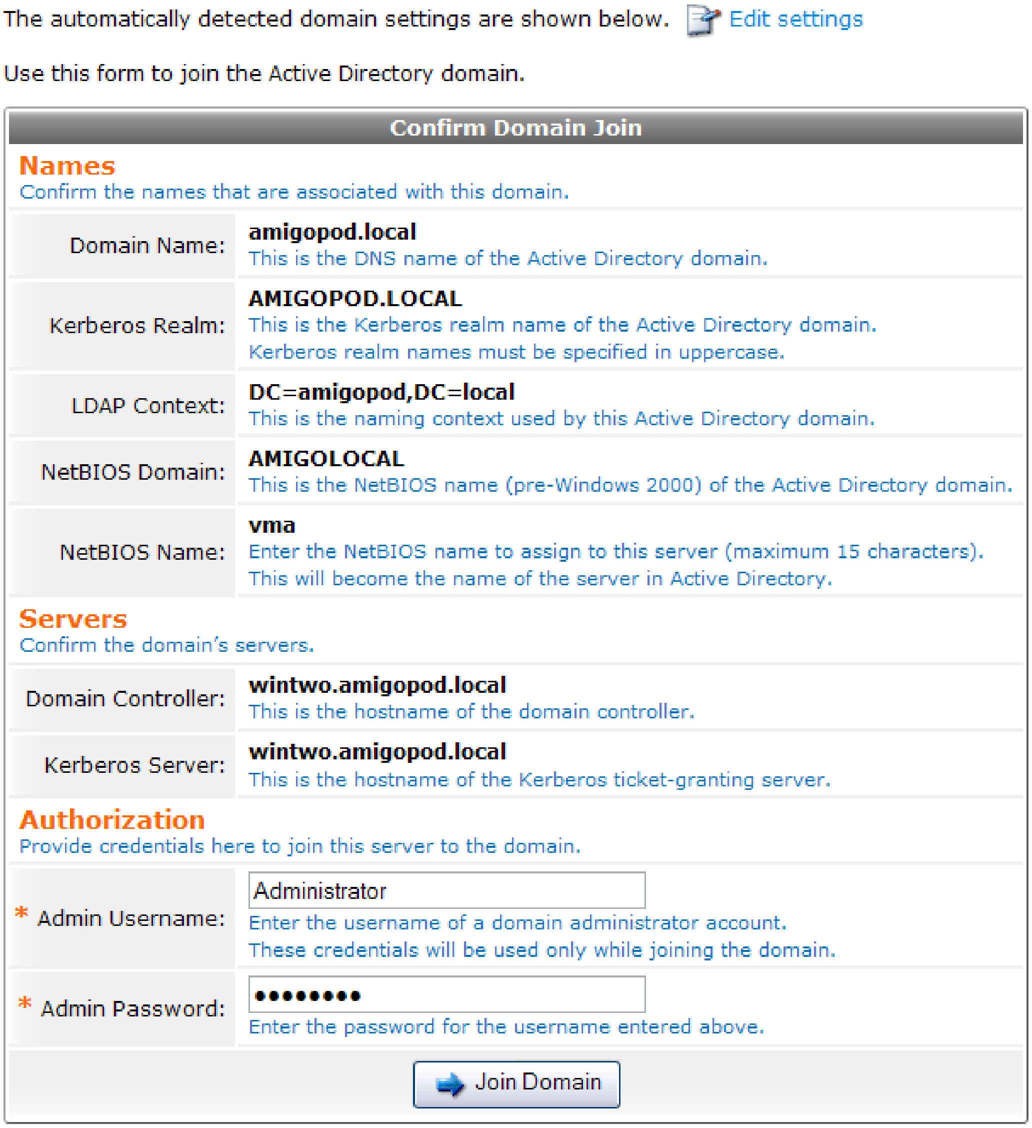
Check the system's hostname at System Hostname .

Hint: You might want to set the system's hostname to **amigopod.amigopod.local**.

Join Active Directory Domain	
* Domain Name:	amigopod.local Enter the DNS name of the Active Directory domain.

* required field

When the server's DNS and network settings are correctly configured, all the necessary domain-related information is automatically detected:



The automatically detected domain settings are shown below. Edit settings

Use this form to join the Active Directory domain.

Confirm Domain Join

Names

Confirm the names that are associated with this domain.

Domain Name:	**amigopod.local** This is the DNS name of the Active Directory domain.
Kerberos Realm:	**AMIGOPOD.LOCAL** This is the Kerberos realm name of the Active Directory domain. Kerberos realm names must be specified in uppercase.
LDAP Context:	**DC=amigopod,DC=local** This is the naming context used by this Active Directory domain.
NetBIOS Domain:	**AMIGOLOCAL** This is the NetBIOS name (pre-Windows 2000) of the Active Directory domain.
NetBIOS Name:	**vma** Enter the NetBIOS name to assign to this server (maximum 15 characters). This will become the name of the server in Active Directory.

Servers

Confirm the domain's servers.

Domain Controller:	**wintwo.amigopod.local** This is the hostname of the domain controller.
Kerberos Server:	**wintwo.amigopod.local** This is the hostname of the Kerberos ticket-granting server.

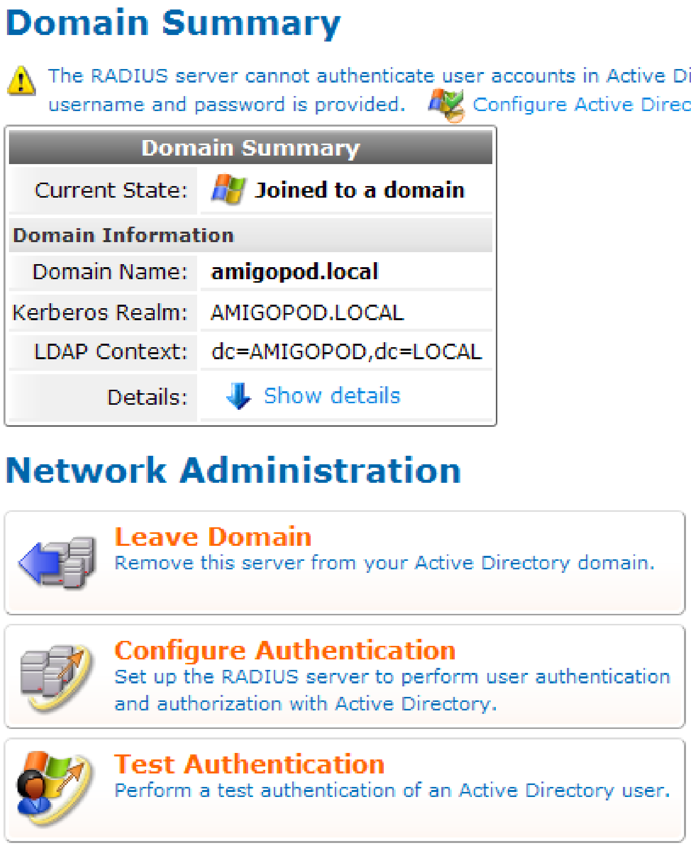
Authorization

Provide credentials here to join this server to the domain.

* Admin Username:	Administrator Enter the username of a domain administrator account. These credentials will be used only while joining the domain.
* Admin Password:	•••••••• Enter the password for the username entered above.

Joining the server to the Active Directory domain then requires entering the username and password for a domain administrator account.

Once the domain has been joined, the status is available on the Active Directory Services page.



Domain Summary

⚠ The RADIUS server cannot authenticate user accounts in Active Directory until a domain username and password is provided. Configure Active Directory authentication

Domain Summary	
Current State:	**Joined to a domain**
Domain Information	
Domain Name:	**amigopod.local**
Kerberos Realm:	AMIGOPOD.LOCAL
LDAP Context:	dc=AMIGOPOD,dc=LOCAL
Details:	Show details

Network Administration

Leave Domain
Remove this server from your Active Directory domain.

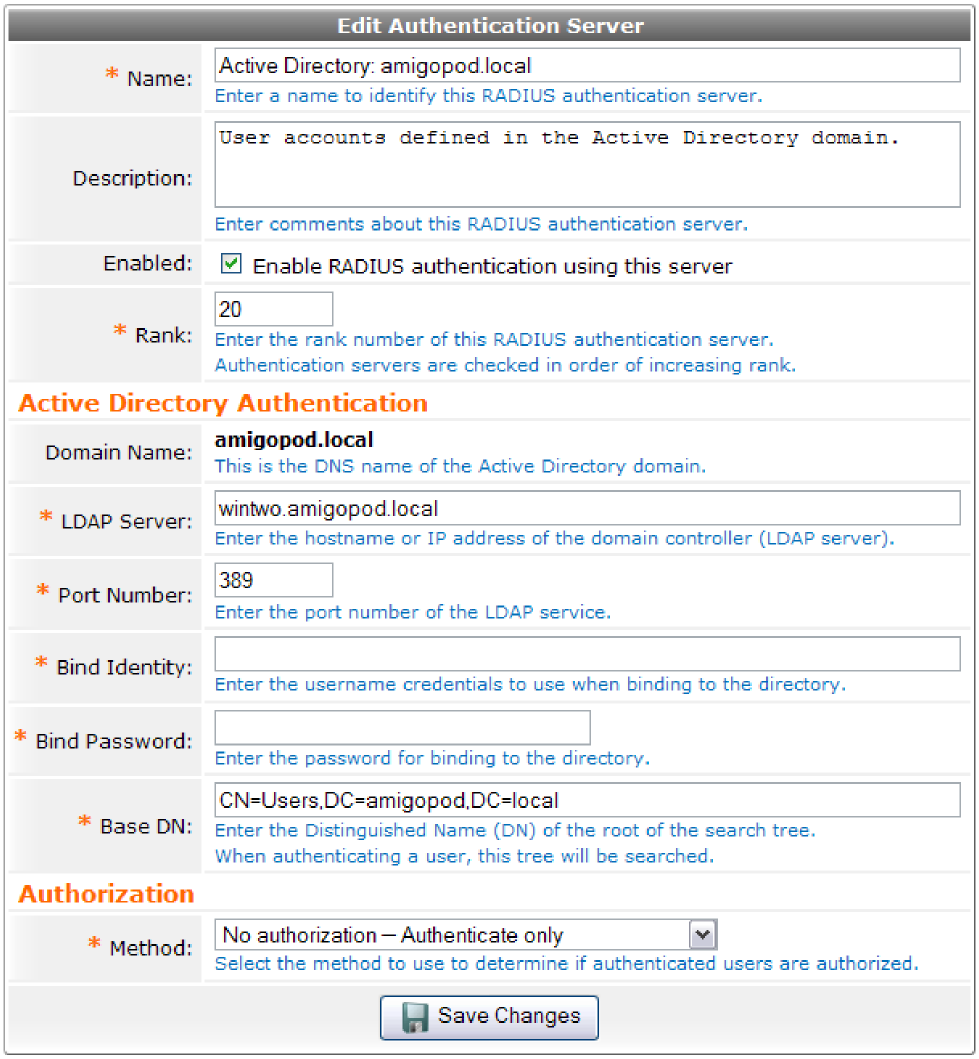
Configure Authentication
Set up the RADIUS server to perform user authentication and authorization with Active Directory.

Test Authentication
Perform a test authentication of an Active Directory user.

Authenticating Active Directory users

As indicated in the domain summary, the RADIUS server cannot authenticate user accounts in Active Directory until a domain username and password is provided.

Clicking the **Configure Authentication** command link displays the Edit Authentication Server form for Active Directory:



Edit Authentication Server

* Name:	Active Directory: amigopod.local Enter a name to identify this RADIUS authentication server.
Description:	User accounts defined in the Active Directory domain. Enter comments about this RADIUS authentication server.
Enabled:	☑ Enable RADIUS authentication using this server
* Rank:	20 Enter the rank number of this RADIUS authentication server. Authentication servers are checked in order of increasing rank.
Active Directory Authentication	
Domain Name:	**amigopod.local** This is the DNS name of the Active Directory domain.
* LDAP Server:	wintwo.amigopod.local Enter the hostname or IP address of the domain controller (LDAP server).
* Port Number:	389 Enter the port number of the LDAP service.
* Bind Identity:	Enter the username credentials to use when binding to the directory.
* Bind Password:	Enter the password for binding to the directory.
* Base DN:	CN=Users,DC=amigopod,DC=local Enter the Distinguished Name (DN) of the root of the search tree. When authenticating a user, this tree will be searched.
Authorization	
* Method:	No authorization — Authenticate only Select the method to use to determine if authenticated users are authorized.

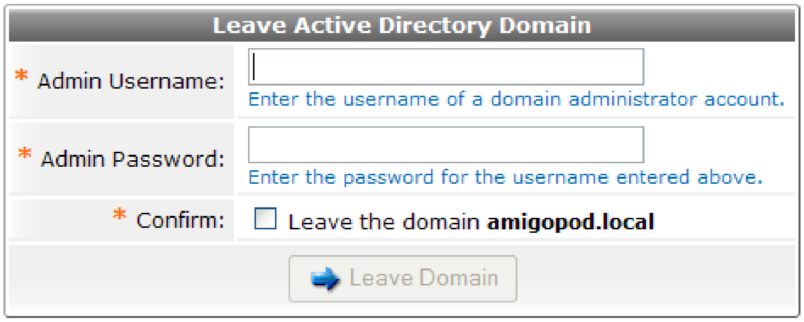
Most of the settings for the authentication server are automatically detected, however a Bind Identity (username) and Bind Password are required in order to authenticate users against the directory.

NOTE The credentials provided do not need to be those of a domain administrator; a restricted user account may be provided here. Only user lookup operations are performed with this user account.

Click the **Save Changes** button to store the credentials for the authentication server.

Leaving an Active Directory domain

The **Leave Domain** command link can be used to remove the server from the domain.



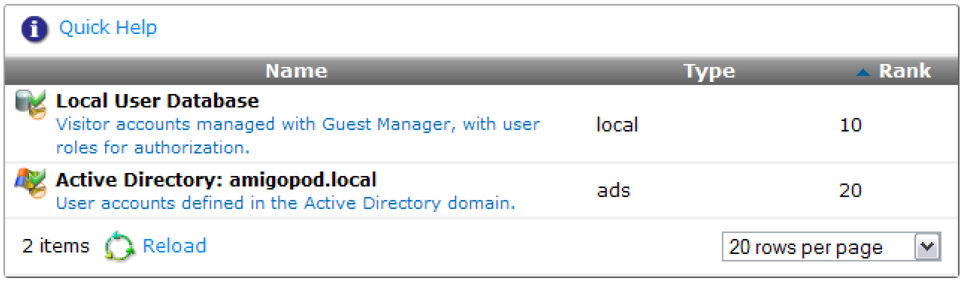
Leave Active Directory Domain

* Admin Username:	Enter the username of a domain administrator account.
* Admin Password:	Enter the password for the username entered above.
* Confirm:	□ Leave the domain **amigopod.local**

As with joining the domain, the credentials for a domain administrator are required to perform this operation.

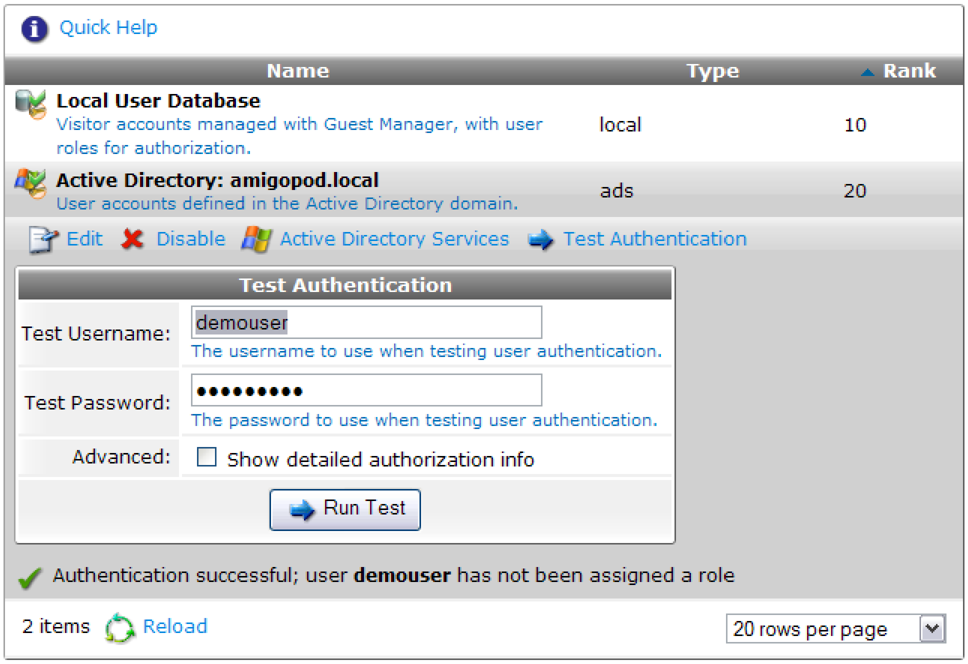
Managing Authentication Servers

The RADIUS Authentication Servers page lists all available sources for use with authentication:



Name	Type	Rank
Local User Database Visitor accounts managed with Guest Manager, with user roles for authorization.	local	10
Active Directory: amigopod.local User accounts defined in the Active Directory domain.	ads	20

The **Test Authentication** command may be used to check the connection to an authentication server, or verify the authorization rules that have been configured:



Name	Type	Rank
Local User Database Visitor accounts managed with Guest Manager, with user roles for authorization.	local	10
Active Directory: amigopod.local User accounts defined in the Active Directory domain.	ads	20

Test Authentication

Test Username:	demouser The username to use when testing user authentication.
Test Password:	••••••••• The password to use when testing user authentication.
Advanced:	□ Show detailed authorization info

✓ Authentication successful; user **demouser** has not been assigned a role

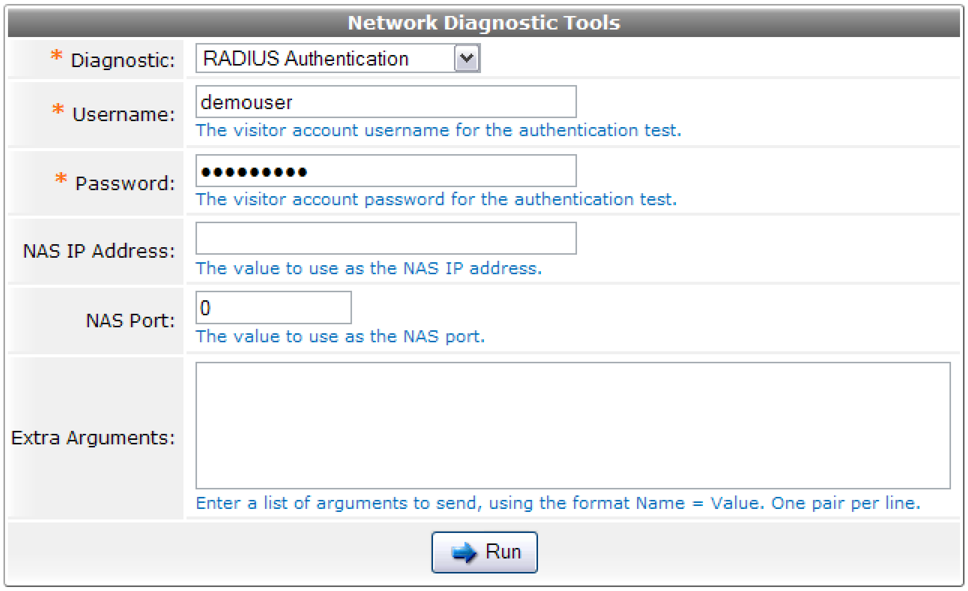
NOTE Changing the properties of an authentication server requires restarting the RADIUS server. When this is necessary, a link is displayed at the top of the page.

Authorization for External Authentication Servers

When a RADIUS Access-Request for a particular user is handled using an external authentication server, the user's authorization is determined by the Authorization settings for that server.

The RADIUS Authentication diagnostic can be used to demonstrate the difference between the various authorization methods.

To use the diagnostic, navigate to **RADIUS** > **Server Control** and click the **Test RADIUS Authentication** command link. Enter the username and password for a user that is externally authenticated.



Network Diagnostic Tools

* Diagnostic: RADIUS Authentication

* Username: demouser
The visitor account username for the authentication test.

* Password: •••••••••
The visitor account password for the authentication test.

NAS IP Address:
The value to use as the NAS IP address.

NAS Port: 0
The value to use as the NAS port.

Extra Arguments:
Enter a list of arguments to send, using the format Name = Value. One pair per line.

Click the **Run** button to perform RADIUS authentication and display the results:

- With authorization method No authorization – Authenticate only:

```
Sending Access-Request of id 165 to 127.0.0.1 port 1812
        User-Name = "demouser"
        User-Password = "XXXXXXXX"
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=165,
length=20
```

Note that in this case, no RADIUS attributes are returned. The Access-Accept or Access-Reject result indicates whether the user was successfully authenticated.

- With authorization method **Assign a fixed user role**:

```
Sending Access-Request of id 122 to 127.0.0.1 port 1812
        User-Name = "demouser"
        User-Password = "XXXXXXXX"
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=122,
length=27
Reply-Message = "Guest"
```

Note that in this case, the RADIUS attribute returned (Reply-Message) corresponds to the user role selected.

- With authorization method **Use PHP code to assign a user role (Advanced)** – more complex authorization rules can be implemented to specify which role to assign to an authenticated user. Authorization can use any of the available properties of the user account, as well as taking into account other factors such as the time of day, previous usage, and more.

Advanced authorization — Example 1

This example covers the case where a domain contains several organizational units (OUs), and the users in each OU are to be mapped to a specific RADIUS role ID.

NOTE

To determine the appropriate role ID, navigate to **RADIUS Services** > **User Roles** and check the ID column for the appropriate role.

For example, to implement the following configuration:

- OU East should be mapped to RADIUS role ID 4
- OU Central should be mapped to RADIUS role ID 5
- OU **West** should be mapped to RADIUS role ID 6

Make sure the following configuration is set:

1. First, ensure that the Base DN for the authentication server is set to the root of the domain – for example: DC=amigopod,DC=local – rather than the "users" container. This is necessary as the organizational units are located below the top level of the directory and cannot be searched from the CN=Users container.
2. Select the authorization method Use PHP code to assign a user role (Advanced) and use the following code:

```
if (stripos($user['distinguishedname'],'OU=East')) return 4;
if (stripos($user['distinguishedname'],'OU=Central')) return 5;
if (stripos($user['distinguishedname'],'OU=West')) return 6;
return false;
```

Explanation: During user authorization, the distinguished name of the user (which will contain the user's OU) is checked against the defined rules, and an appropriate role ID is returned. If no match is found, false is returned, which means that authorization fails and the user's Access-Request will be rejected.

Advanced authorization — Example 2

This example covers the case where users are assigned group memeberships, and users in a particular group are to be mapped to a specific RADIUS role ID.

NOTE

To determine the appropriate role ID, navigate to **RADIUS Services** > **User Roles** and check the ID column for the appropriate role.

For example, to implement the following configuration:

- Members of the Domain Admins group should be mapped to RADIUS role ID 4
- Members of the Users group should be mapped to RADIUS role ID 5
- All other users should be rejected

Make sure the following configuration is set:

- Select the authorization method **Use PHP code to assign a user role (Advanced)** and use the following code:

```
if (in_array('CN=Domain Admins,CN=Users,DC=amigopod,DC=local',
$user['memberof'])) return 4;
if (in_array('CN=Users,CN=Builtin,DC=amigopod,DC=local',
$user['memberof'])) return 5;
return false;
```

Explanation: During user authorization, the 'memberOf' attribute of the user (which will contain a list of the groups to which the user belongs) is checked against the defined rules, and an appropriate role ID is returned. If no match is found, false is returned, which means that authorization fails and the user's Access-Request will be rejected.

NOTE The `in_array()` comparison is done in a case-sensitive manner. Be sure to use the correct case as returned by the LDAP query for the group name. Also note that the complete distinguished name (DN) for the group must be specified, as this is the value checked for in the array of values returned for the 'memberOf' attribute.

NOTE The primary group of a user assigned in Active Directory cannot be checked in this way, as Active Directory does not return the primary group in the values of the 'memberOf' attribute. You can build logic that uses the `$user['primarygroupid']` property instead to work around this issue.