Users, Roles and Orgs - how does it work?

Introduction

Open-AudIT has a granular permissions system to determine what a user within Open-AudIT can do, and the items he can do it to. Open-AudIT can be entirely self-contained, or use Active Directory or OpenLDAP for authentication and/or authorization.

It's entirely up to the administrator of Open-AudIT how they would like the Role Based Access Control to work.

How Does It Work?

A person has an account in the Open-AudIT application. Their user account has a list of associated Roles and Organizations. The roles the user has determines WHAT they can do. The Organizations a user has determines WHICH items they can act upon.

When a user requests to perform an operation (create, read, update, delete) on a collection item, the roles are consulted to see if they are allowed to perform that action, then the orgs are consulted to determine if the collection item belongs to an org the user has permission to act on.

Roles

Open-AudIT ships with inbuilt roles for admin, org_admin and user.

Generally, a user who is an administrator of the Open-AudIT application itself should have admin and possible org_admin roles.

A user can have multiple roles. The permission will be applied at the most permissive level - IE, if a user has the roles of user and org_admin, they will be able to create locations because org_admin grants this permission, even though the user role does not.

The admin role allows access to collections such as configuration, database, groups, ldap servers, logs, queries and roles. Global items that affect the entire application.

The org_admin role usually allows create, read, update and delete actions for any collection that contains the org_id column. Virtually all data except some of the collections mentioned above will contain an org_id column.

The user role generally allows read only access to all items with an org_id column.

Orgs

A user will have a list of associated organizations (orgs). Each org the user has will allow them to act upon items within that org as per their role(s).

All orgs except the default org have a parent. Think of an Org Chart. If a user has permission on an Org, they also have permission on any descendants of that Org.

As at 3.3.2 we have also allowed a user with permission on a child org to see the items from parent orgs for certain collections. Those are: dashboards, discovery_scan_options, fields, files, groups, queries, reports, roles, rules, scripts, summaries, widgets.

Don't forget you have granular control over what users can see and do using Roles in Enterprise.

Their OrgIDs and any descendants	Their OrgIDs only	Their OrgIDs, descendants and ascendants

applications	configuration	dashboards
baselines	database	discovery_scan_options
baselines_policies	errors	fields
buildings	help	files
clouds	nmis	groups
clusters	san	queries
collectors	test	reports
connections	util	roles
credentials		rules
devices		scripts
discoveries		summaries
discovery_log		widgets
floors		
integrations		
ldap_servers		
licenses		
locations		
logs		
networks		
orgs		
rack_devices		
racks		
rooms		
rows		
search		
tasks		
users		

Active Directory and OpenLDAP

Both forms of LDAP can be used for user authentication (is the users name and password correct) as well as user authorization (what roles and orgs does a user have).

If a user is not in the configured LDAP but is in Open-AudIT (eg: the 'admin' user), Open-AudIT will fallback to using itself for both authentication and authorization.

Open-AudIT uses specific LDAP groups for roles and orgs. A user must be a direct member of these group(s) in order for Open-AudIT to determine that users access.

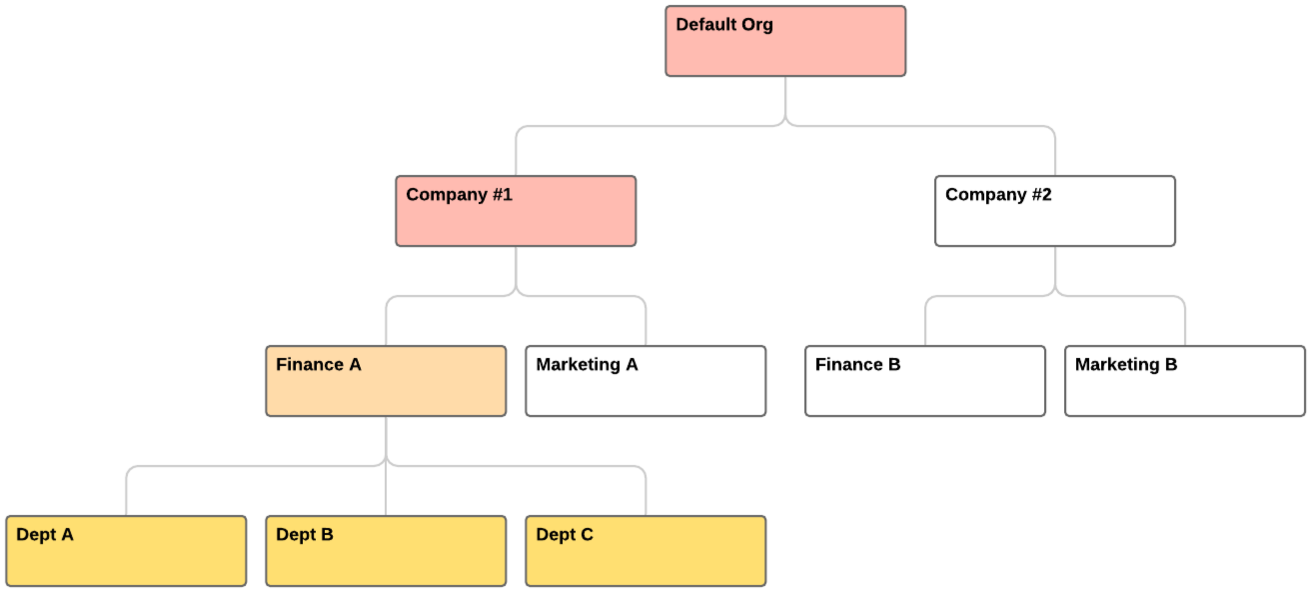
When configured correctly, LDAP use can completely remove the need to create users in Open-AudIT. Simply configure Open-AudIT to use LDAP for both authentication and authorization. If the user does not exist in Open-AudIT but does exist in LDAP and their credentials are correct and they are a member of the required groups Open-AudIT will create the user account automatically.

Example Org Chart with Access

Below you can see an example Org Chart. If a user has permission on the "Finance A" Org, they also have permission on the descendant Orgs of Dept A, B & C. This is regardless of the collection requested.

If the collection requested allows ascendants, then the user will also have access to Company #1 and Default Org items. This is for (as above) queries, groups, et al.

Note - A user may have access to a query from Default Org, but that is the query itself not the result. The result will only show devices that the user has access to - IE devices from Finance A and Dept A, B & C.


Default Org
Company #1
Company #2
Finance A
Marketing A
Finance B
Marketing B
Dept A
Dept B
Dept C