How to configure LDAP

WAS THIS PAGE HELPFUL? Leave Feedback

CONTENTS

Overview

This page reviews AD and LDAP configuration and integration for GroundWork Monitor.

GroundWork Monitor supports single sign-on authentication through an external authentication source. Authentication services can be provided by Microsoft Active Directory or through a standards-compliant LDAP server. Most user accounts need not be defined in GroundWork Monitor a priori. User accounts must still be assigned system-specific roles and privileges, and the use of LDAP for authentication changes the way this is done. Configuring the GroundWork JBoss Portal for LDAP allows user passwords and other details such as role membership to be managed by the external directory service. User accounts and roles are synchronized with the GroundWork JBoss Portal when the user logs in, and users are assigned a default role as well as any roles they are members of in LDAP.

It is also possible to enable LDAPS, or LDAP over SSL, as well as many other alternate configurations. Please see the reference materials for JBoss LDAP configuration available here if you would like to study the various options and customize your LDAP setup with GroundWork Monitor.

This how-to goes through an example of setup, configuration, and assignment of users to roles in the context of users and groups that are managed by LDAP. The following sections outline some important points before you start, requirements and options, and then describes an example of configuring GroundWork Monitor for LDAP authentication.

LDAP and Active Directory Configuration

Important Points Before You Start

- LDAP users cannot be assigned to roles using the portal administrator application.
- LDAP users do not need to be defined in the portal
- Configuration of LDAP parameters is done outside of the UI and requires a restart of `gwservices`.
- User passwords are never synchronized from LDAP to the GroundWork Monitor portal database.
- Any LDAP user who needs to access the GroundWork Monitor portal needs to be part of Authenticated group in LDAP/AD in addition to the regular GroundWork Monitor groups such as `GWAdmin`, `GWOperator` and `GWUser`. If not, the user may get an unauthorized HTTP 403 error when they try to login to GroundWork Monitor.
- GroundWork Monitor must be configured with the LDAP server name, a user name and password to bind to LDAP, and the specific container or organizational unit containing the users to be allowed access.
- The bind user will need browse and search permissions for the locations in LDAP where the users and groups are stored.
- These values must be input into an XML configuration file in a specific location, and the GroundWork Monitor portal must be restarted for the configuration to take effect.
- If your LDAP server is down for some reason, you can revert the GroundWork authentication mechanism back to the GroundWork database. In this case, stock user accounts can be used to login to the portal. Also the passwords for stock user accounts will be the last changed password before switching to LDAP. You would edit the file:

  ```
  /usr/local/groundwork/josso-1.8.4/lib/josso-gateway-config.xml
  ```

  and replace:

  ```
  <s:import resource="josso-gateway-ldap-stores.xml" />
  ```

  with:

  ```
  <s:import resource="josso-gateway-gatein-stores.xml" />
  ```

Requirements and Options

- (Required) Active Directory domain controller or other LDAP provider to which you have administrative access
- (Required) Account with rights to browse the container in which you store the users. Example `ldapauth`, context:

  ```
  cn=ldapauth,ou=GWUsers,dc=demo,dc=com
  ```

> ⚠ Avoid spaces or slashes in the distinguished name of the bind account, and $ or # in the bind account password. These characters can result in situations difficult to troubleshoot where bind attempts from GroundWork fail but with a command line tests succeed.

> ⚠ In at least some scenarios a space in a name of an account or role can be replaced with the string "\u0020". Example follows

```
Represent Joseph Smith as
Joseph\u0020Smith
Represent International Space Station as
International\u0020Space\u0020Station
```

- (Optional) Roles in the portal for desired access levels
- (Optional) A container and groups set up to match roles in the portal

> ⚠ New in the 7.2.1 Rollup patch is the capability to map arbitrarily named LDAP Groups to the standard GW Groups and Roles

- (Useful) adsiedit.msc utility
- Portal Proxy User - The portal proxy user is used to access API's of applications secured by the portal. You can either use an existing user or create a new one. If you do change from the default, you must add this user to the LDAP system you are using to synchronize from. Make sure that the user is member of `GWUser`. The login ID and password in LDAP always has to match the entries for the proxy user settings in the `/usr/local/groundwork/config/foundation.properties` file. You should restart `gwservices` if you modify this file.

```
portal.proxy.user=user
portal.proxy.password=password
```

- GDMA Auto Register User - The GDMA Auto Register user is used to access the Foundation API. It is recommended the default credentials be changed on the GroundWork server before an client agent installation. The login ID and password must match the entries for the `Auto_Register_User` and `Auto_Register_Pass` settings in the GDMA client's `gdma_auto.conf` file. See the Configuring Auto Registration document for adding/changing GDMA credentials.
- Recommended LDAP Setup - Portal access authentication is controlled by *Role* permissions. Users gain access to different sections of the portal through role membership. In LDAP, group membership is used to map Role membership in GroundWork Monitor. In order to manage roles in GroundWork Monitor you should setup a new context (an Organizational Unit in AD by default) for the GroundWork Monitor roles.

  Example: GWRoles (OU=GWRoles)

  To this monitoring specific context add the following *Groups*. With this setup only GroundWork monitoring specific roles will be synchronized even if the user is member of other groups in different contexts (containers). This avoids crowding the GroundWork Monitor administration pages with roles unrelated to monitoring. You can then add company specific groups to the `GWRoles` context, which you can use to define specific access rights in GroundWork Monitor. When a user logs in, *Groups* in the `GWRoles` context of which the user is member of will be synchronized with *Roles* in the portal.
    - GWRoot
    - GWAdmin
    - GWOperator
    - GWUser

Configuration

For LDAP support, "josso-gateway-config.xml" must reference "josso-gateway-ldap-stores.xml" and not "josso-gateway-gatein-stores.xml".

Application on a system previously configured with Josso Ldap store will result in continued operation using the legacy credentials in the "josso-gateway-ldap-stores.xml" file.

To take advantage of the new facilities you can make changes to "foundation.properties". If ldap configurations are found in this file, configuration settings in "josso-gateway-ldap-stores.xml" are ignored.

Details for updating these files are below.

Enabling LDAP Authentication

The use of the LDAP authentication is controlled by the same setting in "josso-gateway-config.xml" which is defaulted to use the local gatein store. To facilitate use of LDAP AD or OpenLDAP you will first change the following line. If LDAP was previously enabled the change will already be present.

Edit "josso-gateway-config.xml" and replace:

/usr/local/groundwork/foundation/container/josso-1.8.4/lib/josso-gateway-config.xml - line 109

```
<s:import resource="josso-gateway-gatein-stores.xml" />
```

with

/usr/local/groundwork/foundation/container/josso-1.8.4/lib/josso-gateway-config.xml - line 109

```
<s:import resource="josso-gateway-ldap-stores.xml" />
```

Endpoint Definitions

The endpoints are added to the foundation.properties file. Each endpoint has a section in the file.

Multiple domains are configured by replicating sets of properties for AD or OpenLDAP below. Only properties that need to be overridden need to be copied, (the rest will default as below based on the type of domain).

- If any domains are configured here, the JOSSO endpoint configuration in josso-gateway-ldap-stores.xml is ignored.
- If NO domains are configured in foundation.properties, the JOSSO configuration is loaded into the LDAP Aggregator as the default domain.

Each specified endpoint is searched separately; the credentials and OU/CN directory in one endpoint have no bearing on the others.

Enabling domain prefixes in usernames

The requirement of using a domain in the user name is controlled by a parameter in foundation.properties whose default is false, no domain required.

/usr/local/groundwork/config/foundation.properties - line 294

```
core.security.ldap.domain_prefix_required = true
```

If multiple endpoint domains are specified and a user name is in more than one, the possibility exists that the authentication will be on the wrong domain and role access will not be granted properly. Therefore we recommend that this be set to true and that users be required to enter the domain string "domain\user".

These are valid login principals (notice the slash can go either way in the login process). The domain name that the user enters is in the example, "demo" or "windows2012":

- `demo\user`
- `windows2012/user`

Domain property naming considerations

Note that domain names in the endpoint definitions have **no relationship to the actual DN domain**. In fact, the domain names these endpoints are known by cannot contain the '.' character. Valid names might be 'Demo' or 'Windows2012'. These generally look like Windows NetBios domain names and are used as prefixes on the principle name during login. So if the actual DN domain name is a simple string you might use it, but observe the rule.

UPN forms are not currently supported. The default domain can also be configured with no domain specified in the properties below. The default domain, if defined, will be used to look up **users that are not authenticated with a domain prefix**.

Otherwise, when a login prefix is not entered for authentication, the named domains are searched **in the order they are defined** in this file and on the first authentication success the search terminates.

Configuring each of the properties for a specific name or default domain must utilize the following forms (core.security.ldap.config.) in the properties file

1. a named domain, (no '.' allowed in domain name):
    a. core.security.ldap.config.<domain name>.<property name> = ...

domain namespaced example

```
core.security.ldap.config.windows2012.provider_url =  ldap://10.0.0.15
```

2. the default domain:
    a. core.security.ldap.config.<property name> = ...

default namespaced example

```
core.security.ldap.config.provider_url =  ldap://10.0.0.15
```

Available LDAP configuration properties

Normally, only these property names need to be specified for each domain endpoint:

- server_type
- provider_url
- security_principal
- security_credential
- users_ctx_dn
- roles_ctx_dn

This is the full list of property names that can be configured per domain:

- credential_query_string
- enable_start_tls
- initial_context_factory
- ldap_search_scope
- principal_uid_attribute_id
- principle_lookup_attribute_id
- provider_url
- role_attribute_id
- role_matching_mode
- roles_ctx_dn
- security_authentication
- security_credential
- security_principal
- security_protocol
- server_type
- trust_store
- trust_store_password
- uid_attribute_id
- updatable_credential_attribute_id
- user_certificate_attribute_id
- user_properties_query_string
- users_ctx_dn

Security credential encryption

The security credential is required to be encrypted. Bare passwords will fail to authenticate. Use the following command lines to generate the string, substituting the actual password for the example PASSWORD

```
/usr/local/groundwork/foundation/scripts/encrypt.sh encrypt --value=PASSWORD
```

The result will look like this:

```
758GJyJCpKEpWZzDsvnfQhJWG9gVw12Tz
```

This value will be used for the security_credential property for the corresponding domain configuration. This procedure is also documented here: How to generate encrypted credentials

Portal super user

The default super user account for configuring defaults and shared dashboards has a username of "root". It is required that the super user account exists in LDAP so it can be logged into the portal, however this username is usually restricted in most organizations. To change the username for the account follow the procedure documented here: How to change the portal super user

LDAP search scope

Where users are in a single master Users OU, the search has only a single level. But suppose the customer has users in a nested form, buried in subdirectories. The Aggregator allows us to define an endpoint at the top of the directory tree, and the search will descend the tree until it has either exhausted the possibilities (no match) or discovered the user (first match) and attempted authentication. The attribute in the endpoint spec that controls this is

```
ldap_search_scope = SUBTREE
```

Alternatively, you may define two or more endpoints for the same LDAP, naming scope as the "BASE" (just the indicated container or OU) or "ONE LEVEL" (objects subordinate to the named base but *not the base*) instead of "SUBTREE" (the base name and all nested objects to the maximum depth). In this way you can limit the searches according to the manner by which the customer has organized users.

LDAPS connections

When connecting to an LDAP provider that is protected by SSL or TLS two changes are needed:

1. Install the certificates from the CA into the certificate store:

   ```
   /usr/local/groundwork/java/bin/keytool keytool -import -noprompt -storepass changeit -keystore
   /usr/local/groundwork/java/jre/lib/security/cacerts -alias MY-CERTIFICATE-NAME -file
   MY-CERTIFICATE-NAME.pem
   ```

   > ✓ **Certificate Encoding**
   > The certificates being imported MUST be PEM encoded certificates or the import will fail. If exporting from Microsoft you should select a Base64 encoded CER certificate type. Do not include the private key when you export.

2. change the protocol used in the `provider_url` from `ldap://` to `ldaps://`:

   ```
   core.security.ldap.config.provider_url = ldaps://10.0.0.35
   ```

Examples

Here are three examples of working configurations.

Take special note of the "domain" portion of the configuration in each example. Both "windows2012" and "demo" are arbitrary. The actual domain names are "corp" and "demo". We suggest avoiding using the actual domain name so that it does not become the unconscious rule, later causing an error where the actual domain had a character like "." embedded.

Whatever you decide, the string you choose will be the one that users must enter, so for example "demo/jdoe" or "windows2012\jsmith". The slash can go either way, the form is always domain followed by slash followed by user.

Note that the port is not specified if the server you are pointing to is using default settings (389 for cert-less, 636 for LDAPS). In the third example you can see the form used for specified port.

```
# 'windows2012' AD endpoint:
#
core.security.ldap.config.windows2012.server_type = AD
core.security.ldap.config.windows2012.provider_url = ldap://10.0.0.15
core.security.ldap.config.windows2012.security_principal = cn=ldapauth,cn=Users,dc=corp,dc=localdomain
core.security.ldap.config.windows2012.security_credential = XcuJVdPmzFo9egZ4a24XFsoTzoeZafKM
core.security.ldap.config.windows2012.users_ctx_dn = cn=Users,dc=corp,dc=localdomain
core.security.ldap.config.windows2012.roles_ctx_dn = ou=GWRoles,dc=corp,dc=localdomain
#
# 'demo' AD endpoint:
#
core.security.ldap.config.demo.server_type = AD
core.security.ldap.config.demo.provider_url = ldaps://10.0.0.25
core.security.ldap.config.demo.security_principal = cn=ldapauth,cn=Users,dc=demo,dc=com
core.security.ldap.config.demo.security_credential = 2eH7t2u82Cc4nfeNqhQfxK3mboEMkMBmY
core.security.ldap.config.demo.users_ctx_dn = cn=Users,dc=demo,dc=com
core.security.ldap.config.demo.roles_ctx_dn = ou=GWRoles,dc=demo,dc=com
#
# 'default' AD endpoint:
#
core.security.ldap.config.server_type = AD
core.security.ldap.config.provider_url = ldaps://10.0.0.35:636
core.security.ldap.config.security_principal = cn=ldapauth,cn=Users,dc=demo,dc=com
core.security.ldap.config.security_credential = 3eH7t2u82Cc4nfeNqW7fxK3mboEMkMBmY
core.security.ldap.config.users_ctx_dn = cn=Users,dc=demo,dc=com
core.security.ldap.config.roles_ctx_dn = ou=GWRoles,dc=demo,dc=com
```

Notes for 7.2.1 Rollup Patch

We recommend that you apply the patch the following patch located here:
https://kb.groundworkopensource.com/display/DOC721/GWME-7.2.1-00+Rollup+Patch+Details

Once this is done you have access to more complex LDAP configuration combinations.

Our recommendation is to use local AD conventions for naming the role context container/ou and the *Groups* therein.

- You can then add users directly to the *Groups*, or add other LDAP groups (which contain users) to the *Groups*.
- All groups used in this scheme **must** be located in one or more role contexts.
- The key thing to remember is that Group membership for any User is the Join of membership in the *Groups* and member groups in each of the role contexts and the User's membership in **one or more** of them.

GroundWork LDAP Aggregator maps these arbitrarily named *Groups* to the standard Groups and Roles already present in our portal. If you add your own Roles those are similarly mapped.

Let's imagine that in your LDAP directory you created:

- a top level OU called **NotGWRoles**
- and in it you created 4 groups:
  - NotGWRoot
  - NotGWAdmin
  - NotGWOperator
  - NotGWUser

Let's say you already have groups directly in the Users tree with users in them that would get the Admin and Operator privileges and these are in your directory as follows:

- administrator users are in "aga"
- operator users are in "ago"

So you make:

- aga a memberof the NotGWAdmin group above
- ago a memberof the NotGWOperator group above

Now on the GW Monitor server add this directive to the file `/usr/local/groundwork/config/foundation.properties`:

```
# Whether LDAP mapping is enabled (default is false)
core.security.ldap.mapping_enabled = true
```

Since you have multiple role context locations in LDAP, you can specify to search each one with separate definitions on the same line separated by a "|" (pipe) symbol. An example follows, directing the LDAP Aggregator to search for groups in each of them and for the User as a member of at least one of them:

```
core.security.ldap.config.atlas.roles_ctx_dn =
ou=NotGWRoles,dc=company,dc=com|cn=Users,dc=company,dc=com
```

If you have Users in multiple places in the directory the same notation can be used for the users context. In the example we imagine there are two sets of users who will be given access, in the NOC and IT groups:

```
core.security.ldap.config.atlas.users_ctx_dn =
cn=NOC,Users,dc=company,dc=com|cn=IT,cn=Users,dc=company,dc=com
```

Add the mapping to the file `/usr/local/groundwork/config/ldap-mapping-directives.properties`. This is where the arbitrary LDAP names get associated with the portal:

```
NotGWRoot=GWRoot
NotGWAdmin=GWAdmin
NotGWOperator=GWOperator
NotGWUser=GWUser
```

Notice there is no mention of the "chained" groups aga and ago. The ldap search conducted by the LDAP Aggregator will match up the combinations and identify the ultimate group and role to be assigned to the user.

Once these changes are made and saved, restart GroundWork services:

```
/etc/init.d/groundwork restart gwservices
```

This covers the 4 standard out of the box groups and roles. Should you want to customize with new roles you must first create the Role (membership) in the portal and assign the host group and resource privileges it may attain. You should also make the adjustments to the pages, containers and application permissions so that the new Role is permitted to access them.

Next you must add the key/value pairs to the ldap-mapping-directives.properties file to map the LDAP group to the portal role. Let's say you named the Role "Management":

```
SomeLDAPGroup=Management
```

Restart gwservices to make the change effective.

On the LDAP side, create the group "SomeLDAPGroup" under one of the named role context locations (from our example, NotGWRoles) and make the Users who will have this privilege members of SomeLDAPGroup. If they are not already members of the corresponding LDAP group for the gw-portal-user role, make that change as well (so they are able to land on the default landing page; if you don't do this their log in will net a 403 not authorized error)

Troubleshooting

If you need more detailed messages in the `/usr/local/groundwork/jpp/standalone/log/framework.log` log file you can increase the logging level. Consult with GroundWork Support to obtain this direction.