SIMIAN systems

Sitellite LDAP Administrator Guide

Sitellite Enterprise Edition

Environment

In order for the Sitellite LDAP driver to work, PHP must be compiled with its LDAP extension enabled. Instructions on installing and configuring the PHP LDAP extension can be found in the PHP manual here:

```
http://www.php.net/ldap
```

Settings

In Sitellite you can either replace the user authentication source entirely with your LDAP directory or you can add the LDAP directory as a secondary source. In the latter case, the sources are tried in the order they are specified until a matching user is found or until there are no more sources.

The file that contains the session configuration is `inc/sessions/sitellite/settings.php`. This file is INI-formatted and contains a series of sections pertaining to different aspects of Sitellite's authentication and session handling process. The sections that pertain to your LDAP configuration are named `[Source 1]`, `[Source 2]`, etc.

The first source section looks like this by default:

```
[Source 1]

driver          = Database
database        = db
tablename       = sitellite_user
map username    = username
map password    = password
map sessionid   = session_id
map timeout     = expires
```

Below that, an example LDAP configuration can be found but is commented out. It looks like this:

```
;[Source 2]
;
;driver         = LDAP
;host           = 127.0.0.1
;port           = 389
;rdn            = ""
;password       = ""
;dn             = ""
;map username   = uid
;map password   = userPassword
```

```
;map sessionid  = session_id
;map timeout    = expires
;set role       = member
;set team       = none
```

If uncommented, this section would tell Sitellite to look for an LDAP directory on the same machine as the web server, listening on port 389. A value for the `rdn`, `password`, and `dn` settings would need to be specified before Sitellite would know where to look for users within the LDAP directory.

The rest of the settings do one of two things:

1. `map` - These settings tell Sitellite to map a field in the session source driver to the specified field in the LDAP directory. For example, `map username = uid` maps the Sitellite `username` field to a field named `uid` in your LDAP directory.
2. `set` - These settings tell Sitellite to set a field in the session source driver to the specified value for any user authenticated from this source. For example, `set role = member` causes any user in the LDAP directory to be mapped to the `member` role in Sitellite.

Authentication Techniques

The default `auth` value if unspecified is "rdn", which tells Sitellite to use the value in the `rdn` as the LDAP rdn value, akin to a username value for logging into LDAP to then look up user information.

If this value is changed to "user" then the rdn value sent to LDAP is dynamically generated as a combination of the username and the dn values in the pattern "cn={username},{dn}", where {username} is the username specified from the user attempting to log into Sitellite, and {dn} is the value specified in the `dn` field in your settings file. Changing the `auth` value is a matter of adding the following line to the LDAP section of the settings file:

```
auth = user
```

groupMembership

Sitellite uses the `groupMembership` field in LDAP to map users to specific roles and teams. However, if your LDAP directory uses a different name for the equivalent field, you can specify an alternate name by adding the following line to the LDAP section of the settings file:

```
groupMembership = alternate_field_name
```

Multiple Contexts

In your LDAP configuration you may have user accounts you wish to allow access to Sitellite in different locations within your LDAP directory tree. Sitellite allows you to specify additional locations by specifying a list of multiple `dn` values in your LDAP settings as follows:

```
dn 1 = "ou=hr,o=org_name"
dn 2 = "ou=sales,o=org_name"
dn 2 = "ou=dev,o=org_name"
```

This would tell Sitellite to check each of the `hr`, `sales`, and `dev` locations in your LDAP directory, instead of just the one specified in your `dn` line. Note that the numbered `dn` values replace the `dn` value entirely, so the latter is no longer needed.

Secure LDAP Connections

To enable a TLS-encrypted secure connection between Sitellite and the LDAP server, simply add the following line to the LDAP section of the settings file:

```
secure = yes
```

Please note that this is not the same as `ldaps`, which is a deprecated encryption technique that has been replaced with LDAP over TLS. The noticeable difference is that `ldaps` used port 636, while TLS is able to continue to use the default LDAP port 389.

Role/Team Mapping

In the example above, roles and teams were hard-coded in the settings file for an entire session source. However, Sitellite can also dynamically map roles and teams from fields in the source.

The roles and teams themselves are always defined in Sitellite. For LDAP users however, since the user accounts don't exist in the `sitellite_user` database table as they would for ordinary Sitellite users, it is in the LDAP directory that the roles and teams must be assigned to individual users.

Sitellite will automatically map roles specified in the `cn` value of the `groupMembership` field for a given user in LDAP to the corresponding role name in Sitellite. The role names in LDAP however have to be prefixed with the string "sitellite_" so that the LDAP groupMembership cn value "sitellite_master" will map to the "master" role in Sitellite.

Please note that only one value may be specified for the user roles, so if multiple values are specified for a given user, only the last one returned by LDAP will be used.

Similarly, Sitellite will map groupMembership values that begin with "sitellite_tm_" to corresponding teams in Sitellite. For example, the LDAP groupMembership cn value "sitellite_tm_hr" will map to the "hr" team in Sitellite. The special team name "sitellite_tm_all" can also be used to grant a user access to all teams, as you might for a "master" user.

Please note that unlike the role values, you can specify as many team values for a user as you need. The first value returned by LDAP will be the official team for that user, and additional values will be added as allowed teams in the same way that you can specify additional teams when editing users in the Sitellite control panel.

Using this naming scheme, you can achieve the same level of flexibility that Sitellite's internal user authentication provides using your external LDAP authentication source.

LDAP Performance

Sitellite's LDAP driver caches information from the LDAP source after a user has successfully logged in, so that the LDAP connection needs only be made once per successful authentication attempt. This keeps the LDAP usage to a minimum while still offering a secure and complete LDAP-based user authentication system.

Managing Users

If the LDAP driver is used as the primary session authentication source, the user administration component in Sitellite should no longer be used. The role and team definitions must still be managed in Sitellite, but since the LDAP driver is a read-only session source, adding or editing users in Sitellite will have no effect.

LDAP user accounts for Sitellite users should continue to be managed using your existing LDAP administrative tools.