Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sssd-ldap.5' command

$ man sssd-ldap.5

SSSD-LDAP(5) File Formats and Conventions SSSD-LDAP(5)

NAME

sssd-ldap - SSSD LDAP provider

DESCRIPTION

This manual page describes the configuration of LDAP domains for sssd(8). Refer to the ?FILE FORMAT? section of the sssd.conf(5) manual page for detailed syntax information.

You can configure SSSD to use more than one LDAP domain.

LDAP back end supports id, auth, access and chpass providers. If you want to authenticate against an LDAP server either TLS/SSL or LDAPS is required. sssd does not support authentication over an unencrypted channel. If the LDAP server is used only as an identity provider, an encrypted channel is not needed. Please refer to ?ldap_access_filter? config option for more information about using LDAP as an access provider.

CONFIGURATION OPTIONS

All of the common configuration options that apply to SSSD domains also apply to LDAP domains. Refer to the ?DOMAIN SECTIONS? section of the sssd.conf(5) manual page for full details. Note that SSSD LDAP mapping attributes are described in the sssd-ldap-attributes(5) manual page.

ldap_uri, ldap_backup_uri (string)

Specifies the comma-separated list of URIs of the LDAP servers to which SSSD should connect in the order of preference. Refer to the

?FAILOVER? section for more information on failover and server redundancy. If neither option is specified, service discovery is enabled. For more information, refer to the ?SERVICE DISCOVERY? section.

The format of the URI must match the format defined in RFC 2732:

ldap[s]://<host>[:port]

For explicit IPv6 addresses, <host> must be enclosed in brackets []

example: ldap://[fc00::126:25]:389

ldap_chpass_uri, ldap_chpass_backup_uri (string)

Specifies the comma-separated list of URIs of the LDAP servers to which SSSD should connect in the order of preference to change the password of a user. Refer to the ?FAILOVER? section for more information on failover and server redundancy.

To enable service discovery ldap_chpass_dns_service_name must be set.

Default: empty, i.e. ldap_uri is used.

ldap_search_base (string)

The default base DN to use for performing LDAP user operations.

Starting with SSSD 1.7.0, SSSD supports multiple search bases using the syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree".

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

Examples:

ldap_search_base = dc=example,dc=com (which is equivalent to)

ldap_search_base = dc=example,dc=com?subtree?

ldap_search_base = cn=host_specific,dc=example,dc=com?subtree?(host=thishost)?dc=example.com?subtree?

Note: It is unsupported to have multiple search bases which reference identically-named objects (for example, groups with the same name in two different search bases). This will lead to unpredictable behavior on client machines.

Default: If not set, the value of the defaultNamingContext or namingContexts attribute from the RootDSE of the LDAP server is used. If defaultNamingContext does not exist or has an empty value namingContexts is used. The namingContexts attribute must have a single value with the DN of the search base of the LDAP server to make this work. Multiple values are are not supported.

ldap_schema (string)

Specifies the Schema Type in use on the target LDAP server. Depending on the selected schema, the default attribute names retrieved from the servers may vary. The way that some attributes are handled may also differ.

Four schema types are currently supported:

? rfc2307

? rfc2307bis

? IPA

? AD

The main difference between these schema types is how group memberships are recorded in the server. With rfc2307, group members are listed by name in the memberUid attribute. With rfc2307bis and IPA, group members are listed by DN and stored in the member attribute. The AD schema type sets the attributes to correspond with Active Directory 2008r2 values.

Default: rfc2307

ldap_pwmodify_mode (string)

Specify the operation that is used to modify user password.

Two modes are currently supported:

? exop - Password Modify Extended Operation (RFC 3062)

? ldap_modify - Direct modification of userPassword (not recommended).

Note: First, a new connection is established to verify current password by binding as the user that requested password change. If successful, this connection is used to change the password therefore the user must have write access to userPassword

attribute.

Default: exop

ldap_default_bind_dn (string)

The default bind DN to use for performing LDAP operations.

ldap_default_authtok_type (string)

The type of the authentication token of the default bind DN.

The two mechanisms currently supported are:

password

obfuscated_password

Default: password

See the sss_obfuscate(8) manual page for more information.

ldap_default_authtok (string)

The authentication token of the default bind DN.

ldap_force_upper_case_realm (boolean)

Some directory servers, for example Active Directory, might deliver the realm part of the UPN in lower case, which might cause the authentication to fail. Set this option to a non-zero value if you want to use an upper-case realm.

Default: false

ldap_enumeration_refresh_timeout (integer)

Specifies how many seconds SSSD has to wait before refreshing its cache of enumerated records.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 300

ldap_purge_cache_timeout (integer)

Determine how often to check the cache for inactive entries (such as groups with no members and users who have never logged in) and remove them to save space.

Setting this option to zero will disable the cache cleanup operation. Please note that if enumeration is enabled, the cleanup task is required in order to detect entries removed from the server and can't be disabled. By default, the cleanup task will run every

3 hours with enumeration enabled.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 0 (disabled)

ldap_group_nesting_level (integer)

If ldap_schema is set to a schema format that supports nested groups (e.g. RFC2307bis), then this option controls how many levels of nesting SSSD will follow. This option has no effect on the RFC2307 schema.

Note: This option specifies the guaranteed level of nested groups to be processed for any lookup. However, nested groups beyond this limit may be returned if previous lookups already resolved the deeper nesting levels. Also, subsequent lookups for other groups may enlarge the result set for original lookup if re-queried.

If ldap_group_nesting_level is set to 0 then no nested groups are processed at all. However, when connected to Active-Directory Server 2008 and later using ?id_provider=ad? it is furthermore required to disable usage of Token-Groups by setting ldap_use_tokengroups to false in order to restrict group nesting.

Default: 2

ldap_use_tokengroups

This options enables or disables use of Token-Groups attribute when performing initgroup for users from Active Directory Server 2008 and later.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: True for AD and IPA otherwise False.

ldap_host_search_base (string)

Optional. Use the given string as search base for host objects.

See ?ldap_search_base? for information about configuring multiple search bases.

Default: the value of ldap_search_base

ldap_service_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_iphost_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_ipnetwork_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_search_timeout (integer)

Specifies the timeout (in seconds) that ldap searches are allowed to run before they are cancelled and cached results are returned (and offline mode is entered)

Note: this option is subject to change in future versions of the SSSD. It will likely be replaced at some point by a series of timeouts for specific lookup types.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 6

ldap_enumeration_search_timeout (integer)

Specifies the timeout (in seconds) that ldap searches for user and group enumerations are allowed to run before they are cancelled and cached results are returned (and offline mode is entered)

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 60

ldap_network_timeout (integer)

Specifies the timeout (in seconds) after which the poll(2)/select(2) following a connect(2) returns in case of no activity.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 6

ldap_opt_timeout (integer)

Specifies a timeout (in seconds) after which calls to synchronous LDAP APIs will abort if no response is received. Also controls the timeout when communicating with the KDC in case of SASL bind, the timeout of an LDAP bind operation, password change extended operation and the StartTLS operation.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 8

ldap_connection_expire_timeout (integer)

Specifies a timeout (in seconds) that a connection to an LDAP server will be maintained. After this time, the connection will be re-established. If used in parallel with SASL/GSSAPI, the sooner of the two values (this value vs. the TGT lifetime) will be used.

If the connection is idle (not actively running an operation) within ldap_opt_timeout seconds of expiration, then it will be closed early to ensure that a new query cannot require the connection to remain open past its expiration. This implies that connections will always be closed immediately and will never be reused if ldap_connection_expire_timeout <= ldap_opt_timout

This timeout can be extended of a random value specified by ldap_connection_expire_offset

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 900 (15 minutes)

ldap_connection_expire_offset (integer)

Random offset between 0 and configured value is added to ldap_connection_expire_timeout.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 0

ldap_connection_idle_timeout (integer)

Specifies a timeout (in seconds) that an idle connection to an LDAP server will be maintained. If the connection is idle for more than this time then the connection will be closed.

You can disable this timeout by setting the value to 0.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 900 (15 minutes)

ldap_page_size (integer)

Specify the number of records to retrieve from LDAP in a single request. Some LDAP servers enforce a maximum limit per-request.

Default: 1000

ldap_disable_paging (boolean)

Disable the LDAP paging control. This option should be used if the LDAP server reports that it supports the LDAP paging control in its RootDSE but it is not enabled or does not behave properly.

Example: OpenLDAP servers with the paging control module installed on the server but not enabled will report it in the RootDSE but be unable to use it.

Example: 389 DS has a bug where it can only support a one paging control at a time on a single connection. On busy clients, this can result in some requests being denied.

Default: False

ldap_disable_range_retrieval (boolean)

Disable Active Directory range retrieval.

Active Directory limits the number of members to be retrieved in a single lookup using the MaxValRange policy (which defaults to 1500

members). If a group contains more members, the reply would include an AD-specific range extension. This option disables parsing of the range extension, therefore large groups will appear as having no members.

Default: False

ldap_sasl_minssf (integer)

When communicating with an LDAP server using SASL, specify the minimum security level necessary to establish the connection. The values of this option are defined by OpenLDAP.

Default: Use the system default (usually specified by ldap.conf)

ldap_sasl_maxssf (integer)

When communicating with an LDAP server using SASL, specify the maximal security level necessary to establish the connection. The values of this option are defined by OpenLDAP.

Default: Use the system default (usually specified by ldap.conf)

ldap_deref_threshold (integer)

Specify the number of group members that must be missing from the internal cache in order to trigger a dereference lookup. If less members are missing, they are looked up individually.

You can turn off dereference lookups completely by setting the value to 0. Please note that there are some codepaths in SSSD, like the IPA HBAC provider, that are only implemented using the dereference call, so even with dereference explicitly disabled, those parts will still use dereference if the server supports it and advertises the dereference control in the rootDSE object.

A dereference lookup is a means of fetching all group members in a single LDAP call. Different LDAP servers may implement different dereference methods. The currently supported servers are 389/RHDS, OpenLDAP and Active Directory.

Note: If any of the search bases specifies a search filter, then the dereference lookup performance enhancement will be disabled regardless of this setting.

Default: 10

ldap_ignore_unreadable_references (bool)

Ignore unreadable LDAP entries referenced in group's member attribute. If this parameter is set to false an error will be returned and the operation will fail instead of just ignoring the unreadable entry.

This parameter may be useful when using the AD provider and the computer account that sssd uses to connect to AD does not have access to a particular entry or LDAP sub-tree for security reasons.

Default: False

ldap_tls_reqcert (string)

Specifies what checks to perform on server certificates in a TLS session, if any. It can be specified as one of the following values:

never = The client will not request or check any server certificate.

allow = The server certificate is requested. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, it will be ignored and the session proceeds normally.

try = The server certificate is requested. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, the session is immediately terminated.

demand = The server certificate is requested. If no certificate is provided, or a bad certificate is provided, the session is immediately terminated.

hard = Same as ?demand?

Default: hard

ldap_tls_cacert (string)

Specifies the file that contains certificates for all of the Certificate Authorities that sssd will recognize.

Default: use OpenLDAP defaults, typically in /etc/openldap/ldap.conf

ldap_tls_cacertdir (string)

Specifies the path of a directory that contains Certificate

Authority certificates in separate individual files. Typically the file names need to be the hash of the certificate followed by '.0'. If available, cacertdir_rehash can be used to create the correct names.

Default: use OpenLDAP defaults, typically in /etc/openldap/ldap.conf

ldap_tls_cert (string)

Specifies the file that contains the certificate for the client's key.

Default: not set

ldap_tls_key (string)

Specifies the file that contains the client's key.

Default: not set

ldap_tls_cipher_suite (string)

Specifies acceptable cipher suites. Typically this is a colon separated list. See ldap.conf(5) for format.

Default: use OpenLDAP defaults, typically in /etc/openldap/ldap.conf

ldap_id_use_start_tls (boolean)

Specifies that the id_provider connection must also use tls to protect the channel.

Default: false

ldap_id_mapping (boolean)

Specifies that SSSD should attempt to map user and group IDs from the ldap_user_objectsid and ldap_group_objectsid attributes instead of relying on ldap_user_uid_number and ldap_group_gid_number. Currently this feature supports only ActiveDirectory objectSID mapping.

Default: false

ldap_min_id, ldap_max_id (integer)

In contrast to the SID based ID mapping which is used if ldap_id_mapping is set to true the allowed ID range for ldap_user_uid_number and ldap_group_gid_number is unbound. In a

setup with sub/trusted-domains this might lead to ID collisions. To avoid collisions ldap_min_id and ldap_max_id can be set to restrict the allowed range for the IDs which are read directly from the server. Sub-domains can then pick other ranges to map IDs.

Default: not set (both options are set to 0)

ldap_sasl_mech (string)

Specify the SASL mechanism to use. Currently only GSSAPI and GSS-SPNEGO are tested and supported.

If the backend supports sub-domains the value of ldap_sasl_mech is automatically inherited to the sub-domains. If a different value is needed for a sub-domain it can be overwritten by setting ldap_sasl_mech for this sub-domain explicitly. Please see TRUSTED DOMAIN SECTION in sssd.conf(5) for details.

Default: not set

ldap_sasl_authid (string)

Specify the SASL authorization id to use. When GSSAPI/GSS-SPNEGO are used, this represents the Kerberos principal used for authentication to the directory. This option can either contain the full principal (for example host/myhost@EXAMPLE.COM) or just the principal name (for example host/myhost). By default, the value is not set and the following principals are used:

hostname@REALM

netbiosname$@REALM

host/hostname@REALM

*$@REALM

host/*@REALM

host/*

If none of them are found, the first principal in keytab is returned.

Default: host/hostname@REALM

ldap_sasl_realm (string)

Specify the SASL realm to use. When not specified, this option defaults to the value of krb5_realm. If the ldap_sasl_authid

contains the realm as well, this option is ignored.

Default: the value of krb5_realm.

ldap_sasl_canonicalize (boolean)

If set to true, the LDAP library would perform a reverse lookup to canonicalize the host name during a SASL bind.

Default: false;

ldap_krb5_keytab (string)

Specify the keytab to use when using SASL/GSSAPI/GSS-SPNEGO. This option can be also set per subdomain or inherited via subdomain_inherit.

Default: System keytab, normally /etc/krb5.keytab

ldap_krb5_init_creds (boolean)

Specifies that the id_provider should init Kerberos credentials (TGT). This action is performed only if SASL is used and the mechanism selected is GSSAPI or GSS-SPNEGO.

Default: true

ldap_krb5_ticket_lifetime (integer)

Specifies the lifetime in seconds of the TGT if GSSAPI or GSS-SPNEGO is used.

This option can be also set per subdomain or inherited via subdomain_inherit.

Default: 86400 (24 hours)

krb5_server, krb5_backup_server (string)

Specifies the comma-separated list of IP addresses or hostnames of the Kerberos servers to which SSSD should connect in the order of preference. For more information on failover and server redundancy, see the ?FAILOVER? section. An optional port number (preceded by a colon) may be appended to the addresses or hostnames. If empty, service discovery is enabled - for more information, refer to the ?SERVICE DISCOVERY? section.

When using service discovery for KDC or kpasswd servers, SSSD first searches for DNS entries that specify _udp as the protocol and falls back to _tcp if none are found.

This option was named ?krb5_kdcip? in earlier releases of SSSD.

While the legacy name is recognized for the time being, users are

advised to migrate their config files to use ?krb5_server? instead.

krb5_realm (string)

Specify the Kerberos REALM (for SASL/GSSAPI/GSS-SPNEGO auth).

Default: System defaults, see /etc/krb5.conf

krb5_canonicalize (boolean)

Specifies if the host principal should be canonicalized when

connecting to LDAP server. This feature is available with MIT

Kerberos >= 1.7

Default: false

krb5_use_kdcinfo (boolean)

Specifies if the SSSD should instruct the Kerberos libraries what

realm and which KDCs to use. This option is on by default, if you

disable it, you need to configure the Kerberos library using the

krb5.conf(5) configuration file.

See the sssd_krb5_locator_plugin(8) manual page for more

information on the locator plugin.

Default: true

ldap_pwd_policy (string)

Select the policy to evaluate the password expiration on the client

side. The following values are allowed:

none - No evaluation on the client side. This option cannot disable

server-side password policies.

shadow - Use shadow(5) style attributes to evaluate if the password

has expired. Please see option "ldap_chpass_update_last_change" as

well.

mit_kerberos - Use the attributes used by MIT Kerberos to determine

if the password has expired. Use chpass_provider=krb5 to update

these attributes when the password is changed.

Default: none

Note: if a password policy is configured on server side, it always

takes precedence over policy set with this option.

ldap_referrals (boolean)

Specifies whether automatic referral chasing should be enabled.

Please note that sssd only supports referral chasing when it is compiled with OpenLDAP version 2.4.13 or higher.

Chasing referrals may incur a performance penalty in environments that use them heavily, a notable example is Microsoft Active Directory. If your setup does not in fact require the use of referrals, setting this option to false might bring a noticeable performance improvement. Setting this option to false is therefore recommended in case the SSSD LDAP provider is used together with Microsoft Active Directory as a backend. Even if SSSD would be able to follow the referral to a different AD DC no additional data would be available.

Default: true

ldap_dns_service_name (string)

Specifies the service name to use when service discovery is enabled.

Default: ldap

ldap_chpass_dns_service_name (string)

Specifies the service name to use to find an LDAP server which allows password changes when service discovery is enabled.

Default: not set, i.e. service discovery is disabled

ldap_chpass_update_last_change (bool)

Specifies whether to update the ldap_user_shadow_last_change attribute with days since the Epoch after a password change operation.

It is recommend to set this option explicitly if "ldap_pwd_policy = shadow" is used to let SSSD know if the LDAP server will update shadowLastChange LDAP attribute automatically after a password change or if SSSD has to update it.

Default: False

ldap_access_filter (string)

If using access_provider = ldap and ldap_access_order = filter

(default), this option is mandatory. It specifies an LDAP search filter criteria that must be met for the user to be granted access on this host. If access_provider = ldap, ldap_access_order = filter and this option is not set, it will result in all users being denied access. Use access_provider = permit to change this default behavior. Please note that this filter is applied on the LDAP user entry only and thus filtering based on nested groups may not work (e.g. memberOf attribute on AD entries points only to direct parents). If filtering based on nested groups is required, please see sssd-simple(5).

Example:

```
access_provider = ldap
ldap_access_filter = (employeeType=admin)
```

This example means that access to this host is restricted to users whose employeeType attribute is set to "admin".

Offline caching for this feature is limited to determining whether the user's last online login was granted access permission. If they were granted access during their last login, they will continue to be granted access while offline and vice versa.

Default: Empty

ldap_account_expire_policy (string)

With this option a client side evaluation of access control attributes can be enabled.

Please note that it is always recommended to use server side access control, i.e. the LDAP server should deny the bind request with a suitable error code even if the password is correct.

The following values are allowed:

shadow: use the value of ldap_user_shadow_expire to determine if the account is expired.

ad: use the value of the 32bit field ldap_user_ad_user_account_control and allow access if the second bit is not set. If the attribute is missing access is granted. Also the expiration time of the account is checked.

rhds, ipa, 389ds: use the value of ldap_ns_account_lock to check if access is allowed or not.

nds: the values of ldap_user_nds_login_allowed_time_map, ldap_user_nds_login_disabled and ldap_user_nds_login_expiration_time are used to check if access is allowed. If both attributes are missing access is granted.

This is an experimental feature, please use https://github.com/SSSD/sssd/ to report any issues.

Please note that the ldap_access_order configuration option must include ?expire? in order for the ldap_account_expire_policy option to work.

Default: Empty

ldap_access_order (string)

Comma separated list of access control options. Allowed values are:

filter: use ldap_access_filter

lockout: use account locking. If set, this option denies access in case that ldap attribute 'pwdAccountLockedTime' is present and has value of '000001010000Z'. Please see the option ldap_pwdlockout_dn. Please note that 'access_provider = ldap' must be set for this feature to work.

Please note that this option is superseded by the ?ppolicy? option and might be removed in a future release.

ppolicy: use account locking. If set, this option denies access in case that ldap attribute 'pwdAccountLockedTime' is present and has value of '000001010000Z' or represents any time in the past. The value of the 'pwdAccountLockedTime' attribute must end with 'Z', which denotes the UTC time zone. Other time zones are not currently supported and will result in "access-denied" when users attempt to log in. Please see the option ldap_pwdlockout_dn. Please note that 'access_provider = ldap' must be set for this feature to work.

expire: use ldap_account_expire_policy

pwd_expire_policy_reject, pwd_expire_policy_warn, pwd_expire_policy_renew: These options are useful if users are

interested in being warned that password is about to expire and authentication is based on using a different method than passwords - for example SSH keys.

The difference between these options is the action taken if user password is expired: pwd_expire_policy_reject - user is denied to log in, pwd_expire_policy_warn - user is still able to log in, pwd_expire_policy_renew - user is prompted to change his password immediately.

Note If user password is expired no explicit message is prompted by SSSD.

Please note that 'access_provider = ldap' must be set for this feature to work. Also 'ldap_pwd_policy' must be set to an appropriate password policy.

authorized_service: use the authorizedService attribute to determine access

host: use the host attribute to determine access

rhost: use the rhost attribute to determine whether remote host can access

Please note, rhost field in pam is set by application, it is better to check what the application sends to pam, before enabling this access control option

Default: filter

Please note that it is a configuration error if a value is used more than once.

ldap_pwdlockout_dn (string)

This option specifies the DN of password policy entry on LDAP server. Please note that absence of this option in sssd.conf in case of enabled account lockout checking will yield access denied as ppolicy attributes on LDAP server cannot be checked properly.

Example: cn=ppolicy,ou=policies,dc=example,dc=com

Default: cn=ppolicy,ou=policies,$ldap_search_base

ldap_deref (string)

Specifies how alias dereferencing is done when performing a search.

The following options are allowed:

never: Aliases are never dereferenced.

searching: Aliases are dereferenced in subordinates of the base object, but not in locating the base object of the search.

finding: Aliases are only dereferenced when locating the base object of the search.

always: Aliases are dereferenced both in searching and in locating the base object of the search.

Default: Empty (this is handled as never by the LDAP client libraries)

ldap_rfc2307_fallback_to_local_users (boolean)

Allows to retain local users as members of an LDAP group for servers that use the RFC2307 schema.

In some environments where the RFC2307 schema is used, local users are made members of LDAP groups by adding their names to the memberUid attribute. The self-consistency of the domain is compromised when this is done, so SSSD would normally remove the "missing" users from the cached group memberships as soon as nsswitch tries to fetch information about the user via getpw*() or initgroups() calls.

This option falls back to checking if local users are referenced, and caches them so that later initgroups() calls will augment the local users with the additional LDAP groups.

Default: false

wildcard_limit (integer)

Specifies an upper limit on the number of entries that are downloaded during a wildcard lookup.

At the moment, only the InfoPipe responder supports wildcard lookups.

Default: 1000 (often the size of one page)

ldap_library_debug_level (integer)

Switches on libldap debugging with the given level. The libldap debug messages will be written independent of the general

debug_level.

OpenLDAP uses a bitmap to enable debugging for specific components, -1 will enable full debug output.

Default: 0 (libldap debugging disabled)

SUDO OPTIONS

The detailed instructions for configuration of sudo_provider are in the manual page sssd-sudo(5).

ldap_sudo_full_refresh_interval (integer)

How many seconds SSSD will wait between executing a full refresh of sudo rules (which downloads all rules that are stored on the server).

The value must be greater than ldap_sudo_smart_refresh_interval

You can disable full refresh by setting this option to 0. However, either smart or full refresh must be enabled.

Default: 21600 (6 hours)

ldap_sudo_smart_refresh_interval (integer)

How many seconds SSSD has to wait before executing a smart refresh of sudo rules (which downloads all rules that have USN higher than the highest server USN value that is currently known by SSSD).

If USN attributes are not supported by the server, the modifyTimestamp attribute is used instead.

Note: the highest USN value can be updated by three tasks: 1) By sudo full and smart refresh (if updated rules are found), 2) by enumeration of users and groups (if enabled and updated users or groups are found) and 3) by reconnecting to the server (by default every 15 minutes, see ldap_connection_expire_timeout).

You can disable smart refresh by setting this option to 0. However, either smart or full refresh must be enabled.

Default: 900 (15 minutes)

ldap_sudo_random_offset (integer)

Random offset between 0 and configured value is added to smart and full refresh periods each time the periodic task is scheduled. The value is in seconds.

Note that this random offset is also applied on the first SSSD start which delays the first sudo rules refresh. This prolongs the time when the sudo rules are not available for use.

You can disable this offset by setting the value to 0.

Default: 0 (disabled)

ldap_sudo_use_host_filter (boolean)

If true, SSSD will download only rules that are applicable to this machine (using the IPv4 or IPv6 host/network addresses and hostnames).

Default: true

ldap_sudo_hostnames (string)

Space separated list of hostnames or fully qualified domain names that should be used to filter the rules.

If this option is empty, SSSD will try to discover the hostname and the fully qualified domain name automatically.

If ldap_sudo_use_host_filter is false then this option has no effect.

Default: not specified

ldap_sudo_ip (string)

Space separated list of IPv4 or IPv6 host/network addresses that should be used to filter the rules.

If this option is empty, SSSD will try to discover the addresses automatically.

If ldap_sudo_use_host_filter is false then this option has no effect.

Default: not specified

ldap_sudo_include_netgroups (boolean)

If true then SSSD will download every rule that contains a netgroup in sudoHost attribute.

If ldap_sudo_use_host_filter is false then this option has no effect.

Default: true

ldap_sudo_include_regexp (boolean)

If true then SSSD will download every rule that contains a wildcard in sudoHost attribute.

If ldap_sudo_use_host_filter is false then this option has no effect.

Note

Using wildcard is an operation that is very costly to evaluate on the LDAP server side!

Default: false

This manual page only describes attribute name mapping. For detailed explanation of sudo related attribute semantics, see sudoers.ldap(5)

AUTOFS OPTIONS

Some of the defaults for the parameters below are dependent on the LDAP schema.

ldap_autofs_map_master_name (string)

The name of the automount master map in LDAP.

Default: auto.master

ldap_autofs_map_object_class (string)

The object class of an automount map entry in LDAP.

Default: nisMap (rfc2307, autofs_provider=ad), otherwise automountMap

ldap_autofs_map_name (string)

The name of an automount map entry in LDAP.

Default: nisMapName (rfc2307, autofs_provider=ad), otherwise automountMapName

ldap_autofs_entry_object_class (string)

The object class of an automount entry in LDAP. The entry usually corresponds to a mount point.

Default: nisObject (rfc2307, autofs_provider=ad), otherwise automount

ldap_autofs_entry_key (string)

The key of an automount entry in LDAP. The entry usually corresponds to a mount point.

Default: cn (rfc2307, autofs_provider=ad), otherwise automountKey

ldap_autofs_entry_value (string)

The key of an automount entry in LDAP. The entry usually corresponds to a mount point.

Default: nisMapEntry (rfc2307, autofs_provider=ad), otherwise automountInformation

Please note that the automounter only reads the master map on startup, so if any autofs-related changes are made to the sssd.conf, you typically also need to restart the automounter daemon after restarting the SSSD.

ADVANCED OPTIONS

These options are supported by LDAP domains, but they should be used with caution. Please include them in your configuration only if you know what you are doing.

ldap_netgroup_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_user_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_group_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

Note

If the option ?ldap_use_tokengroups? is enabled, the searches

against Active Directory will not be restricted and return all groups memberships, even with no GID mapping. It is recommended to disable this feature, if group names are not being displayed correctly.

ldap_sudo_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

ldap_autofs_search_base (string)

An optional base DN, search scope and LDAP filter to restrict LDAP searches for this attribute type.

syntax:

search_base[?scope?[filter][?search_base?scope?[filter]]*]

The scope can be one of "base", "onelevel" or "subtree". The scope functions as specified in section 4.5.1.2 of http://tools.ietf.org/html/rfc4511

The filter must be a valid LDAP search filter as specified by http://www.ietf.org/rfc/rfc2254.txt

For examples of this syntax, please refer to the ?ldap_search_base? examples section.

Default: the value of ldap_search_base

Please note that specifying scope or filter is not supported for searches against an Active Directory Server that might yield a large number of results and trigger the Range Retrieval extension in the response.

FAILOVER

The failover feature allows back ends to automatically switch to a different server if the current server fails.

Failover Syntax

The list of servers is given as a comma-separated list; any number of spaces is allowed around the comma. The servers are listed in order of preference. The list can contain any number of servers.

For each failover-enabled config option, two variants exist: primary and backup. The idea is that servers in the primary list are preferred and backup servers are only searched if no primary servers can be reached. If a backup server is selected, a timeout of 31 seconds is set. After this timeout SSSD will periodically try to reconnect to one of the primary servers. If it succeeds, it will replace the current active (backup) server.

The Failover Mechanism

The failover mechanism distinguishes between a machine and a service. The back end first tries to resolve the hostname of a given machine; if this resolution attempt fails, the machine is considered offline. No further attempts are made to connect to this machine for any other service. If the resolution attempt succeeds, the back end tries to connect to a service on this machine. If the service connection attempt fails, then only this particular service is considered offline and the back end automatically switches over to the next service. The machine is still considered online and might still be tried for another service.

Further connection attempts are made to machines or services marked as offline after a specified period of time; this is currently hard coded to 30 seconds.

If there are no more machines to try, the back end as a whole switches to offline mode, and then attempts to reconnect every 30 seconds.

Failover time outs and tuning

Resolving a server to connect to can be as simple as running a single DNS query or can involve several steps, such as finding the correct site or trying out multiple host names in case some of the configured servers are not reachable. The more complex scenarios can take some time and SSSD needs to balance between providing enough time to finish the resolution process but on the other hand, not trying for too long before falling back to offline mode. If the SSSD debug logs show that the server resolution is timing out before a live server is contacted, you can consider changing the time outs.

This section lists the available tunables. Please refer to their description in the sssd.conf(5), manual page.

dns_resolver_server_timeout

Time in milliseconds that sets how long would SSSD talk to a single DNS server before trying next one.

Default: 1000

dns_resolver_op_timeout

Time in seconds to tell how long would SSSD try to resolve single DNS query (e.g. resolution of a hostname or an SRV record) before trying the next hostname or discovery domain.

Default: 3

dns_resolver_timeout

How long would SSSD try to resolve a failover service. This service resolution internally might include several steps, such as resolving DNS SRV queries or locating the site.

Default: 6

For LDAP-based providers, the resolve operation is performed as part of an LDAP connection operation. Therefore, also the ?ldap_opt_timeout? timeout should be set to a larger value than ?dns_resolver_timeout? which in turn should be set to a larger value than ?dns_resolver_op_timeout? which should be larger than

?dns_resolver_server_timeout?.

SERVICE DISCOVERY

The service discovery feature allows back ends to automatically find the appropriate servers to connect to using a special DNS query. This feature is not supported for backup servers.

Configuration

If no servers are specified, the back end automatically uses service discovery to try to find a server. Optionally, the user may choose to use both fixed server addresses and service discovery by inserting a special keyword, ?_srv_?, in the list of servers. The order of preference is maintained. This feature is useful if, for example, the user prefers to use service discovery whenever possible, and fall back to a specific server when no servers can be discovered using DNS.

The domain name

Please refer to the ?dns_discovery_domain? parameter in the sssd.conf(5) manual page for more details.

The protocol

The queries usually specify _tcp as the protocol. Exceptions are documented in respective option description.

See Also

For more information on the service discovery mechanism, refer to RFC 2782.

ID MAPPING

The ID-mapping feature allows SSSD to act as a client of Active Directory without requiring administrators to extend user attributes to support POSIX attributes for user and group identifiers.

NOTE: When ID-mapping is enabled, the uidNumber and gidNumber attributes are ignored. This is to avoid the possibility of conflicts between automatically-assigned and manually-assigned values. If you need to use manually-assigned values, ALL values must be manually-assigned.

Please note that changing the ID mapping related configuration options will cause user and group IDs to change. At the moment, SSSD does not

support changing IDs, so the SSSD database must be removed. Because cached passwords are also stored in the database, removing the database should only be performed while the authentication servers are reachable, otherwise users might get locked out. In order to cache the password, an authentication must be performed. It is not sufficient to use sss_cache(8) to remove the database, rather the process consists of:

? Making sure the remote servers are reachable

? Stopping the SSSD service

? Removing the database

? Starting the SSSD service

Moreover, as the change of IDs might necessitate the adjustment of other system properties such as file and directory ownership, it's advisable to plan ahead and test the ID mapping configuration thoroughly.

Mapping Algorithm

Active Directory provides an objectSID for every user and group object in the directory. This objectSID can be broken up into components that represent the Active Directory domain identity and the relative identifier (RID) of the user or group object.

The SSSD ID-mapping algorithm takes a range of available UIDs and divides it into equally-sized component sections - called "slices"-. Each slice represents the space available to an Active Directory domain.

When a user or group entry for a particular domain is encountered for the first time, the SSSD allocates one of the available slices for that domain. In order to make this slice-assignment repeatable on different client machines, we select the slice based on the following algorithm: The SID string is passed through the murmurhash3 algorithm to convert it to a 32-bit hashed value. We then take the modulus of this value with the total number of available slices to pick the slice.

NOTE: It is possible to encounter collisions in the hash and subsequent modulus. In these situations, we will select the next available slice,

but it may not be possible to reproduce the same exact set of slices on other machines (since the order that they are encountered will determine their slice). In this situation, it is recommended to either switch to using explicit POSIX attributes in Active Directory (disabling ID-mapping) or configure a default domain to guarantee that at least one is always consistent. See ?Configuration? for details.

Configuration

Minimum configuration (in the ?[domain/DOMAINNAME]? section):

ldap_id_mapping = True

ldap_schema = ad

The default configuration results in configuring 10,000 slices, each capable of holding up to 200,000 IDs, starting from 200,000 and going up to 2,000,200,000. This should be sufficient for most deployments.

Advanced Configuration

ldap_idmap_range_min (integer)

Specifies the lower (inclusive) bound of the range of POSIX IDs to use for mapping Active Directory user and group SIDs. It is the first POSIX ID which can be used for the mapping.

NOTE: This option is different from ?min_id? in that ?min_id? acts to filter the output of requests to this domain, whereas this option controls the range of ID assignment. This is a subtle distinction, but the good general advice would be to have ?min_id? be less-than or equal to ?ldap_idmap_range_min?

Default: 200000

ldap_idmap_range_max (integer)

Specifies the upper (exclusive) bound of the range of POSIX IDs to use for mapping Active Directory user and group SIDs. It is the first POSIX ID which cannot be used for the mapping anymore, i.e. one larger than the last one which can be used for the mapping.

NOTE: This option is different from ?max_id? in that ?max_id? acts to filter the output of requests to this domain, whereas this option controls the range of ID assignment. This is a

subtle distinction, but the good general advice would be to have ?max_id? be greater-than or equal to ?ldap_idmap_range_max?

Default: 2000200000

ldap_idmap_range_size (integer)

Specifies the number of IDs available for each slice. If the range size does not divide evenly into the min and max values, it will create as many complete slices as it can.

NOTE: The value of this option must be at least as large as the highest user RID planned for use on the Active Directory server. User lookups and login will fail for any user whose RID is greater than this value.

For example, if your most recently-added Active Directory user has objectSid=S-1-5-21-2153326666-2176343378-3404031434-1107, ?ldap_idmap_range_size? must be at least 1108 as range size is equal to maximal SID minus minimal SID plus one (e.g. 1108 = 1107 - 0 + 1).

It is important to plan ahead for future expansion, as changing this value will result in changing all of the ID mappings on the system, leading to users with different local IDs than they previously had.

Default: 200000

ldap_idmap_default_domain_sid (string)

Specify the domain SID of the default domain. This will guarantee that this domain will always be assigned to slice zero in the ID map, bypassing the murmurhash algorithm described above.

Default: not set

ldap_idmap_default_domain (string)

Specify the name of the default domain.

Default: not set

ldap_idmap_autorid_compat (boolean)

Changes the behavior of the ID-mapping algorithm to behave more

similarly to winbind's ?idmap_autorid? algorithm.

When this option is configured, domains will be allocated starting with slice zero and increasing monatomically with each additional domain.

NOTE: This algorithm is non-deterministic (it depends on the order that users and groups are requested). If this mode is required for compatibility with machines running winbind, it is recommended to also use the ?ldap_idmap_default_domain_sid? option to guarantee that at least one domain is consistently allocated to slice zero.

Default: False

ldap_idmap_helper_table_size (integer)

Maximal number of secondary slices that is tried when performing mapping from UNIX id to SID.

Note: Additional secondary slices might be generated when SID is being mapped to UNIX id and RID part of SID is out of range for secondary slices generated so far. If value of ldap_idmap_helper_table_size is equal to 0 then no additional secondary slices are generated.

Default: 10

Well-Known SIDs

SSSD supports to look up the names of Well-Known SIDs, i.e. SIDs with a special hardcoded meaning. Since the generic users and groups related to those Well-Known SIDs have no equivalent in a Linux/UNIX environment no POSIX IDs are available for those objects.

The SID name space is organized in authorities which can be seen as different domains. The authorities for the Well-Known SIDs are

? Null Authority

? World Authority

? Local Authority

? Creator Authority

? Mandatory Label Authority

? Authentication Authority

? NT Authority

? Built-in

The capitalized version of these names are used as domain names when returning the fully qualified name of a Well-Known SID.

Since some utilities allow to modify SID based access control information with the help of a name instead of using the SID directly SSSD supports to look up the SID by the name as well. To avoid collisions only the fully qualified names can be used to look up Well-Known SIDs. As a result the domain names ?NULL AUTHORITY?, ?WORLD AUTHORITY?, ?LOCAL AUTHORITY?, ?CREATOR AUTHORITY?, ?MANDATORY LABEL AUTHORITY?, ?AUTHENTICATION AUTHORITY?, ?NT AUTHORITY? and ?BUILTIN? should not be used as domain names in sssd.conf.

EXAMPLE

The following example assumes that SSSD is correctly configured and LDAP is set to one of the domains in the [domains] section.

```
[domain/LDAP]
id_provider = ldap
auth_provider = ldap
ldap_uri = ldap://ldap.mydomain.org
ldap_search_base = dc=mydomain,dc=org
ldap_tls_reqcert = demand
cache_credentials = true
```

LDAP ACCESS FILTER EXAMPLE

The following example assumes that SSSD is correctly configured and to use the ldap_access_order=lockout.

```
[domain/LDAP]
id_provider = ldap
auth_provider = ldap
access_provider = ldap
ldap_access_order = lockout
ldap_pwdlockout_dn = cn=ppolicy,ou=policies,dc=mydomain,dc=org
ldap_uri = ldap://ldap.mydomain.org
ldap_search_base = dc=mydomain,dc=org
```

ldap_tls_reqcert = demand

cache_credentials = true

NOTES

The descriptions of some of the configuration options in this manual page are based on the ldap.conf(5) manual page from the OpenLDAP 2.4 distribution.

SEE ALSO

sssd(8), sssd.conf(5), sssd-ldap(5), sssd-ldap-attributes(5), sssd-krb5(5), sssd-simple(5), sssd-ipa(5), sssd-ad(5), sssd-files(5), sssd-sudo(5), sssd-session-recording(5), sss_cache(8), sss_debuglevel(8), sss_obfuscate(8), sss_seed(8), sssd_krb5_locator_plugin(8), sss_ssh_authorizedkeys(8), sss_ssh_knownhostsproxy(8), sssd-ifp(5), pam_sss(8). sss_rpcidmapd(5) sssd-systemtap(5)

AUTHORS

The SSSD upstream - https://github.com/SSSD/sssd/

SSSD 07/10/2023 SSSD-LDAP(5)