NAME

sssd.conf − the configuration file for SSSD

FILE FORMAT

The file has an ini−style syntax and consists of sections and parameters. A section begins with the name of the section in square brackets and continues until the next section begins. An example of section with single and multi−valued parameters:

[section]
key = value
key2 = value2,value3

The data types used are string (no quotes needed), integer and bool (with values of “TRUE/FALSE”).

A line comment starts with a hash sign (“#”) or a semicolon (“;”). Inline comments are not supported.

All sections can have an optional *description* parameter. Its function is only as a label for the section.

sssd.conf must be a regular file, owned by root and only root may read from or write to the file.

GENERAL OPTIONS

Following options are usable in more than one configuration sections.

Options usable in all sections

debug_level (integer)

SSSD supports two representations for specifying the debug level. The simplest is to specify a decimal value from 0−9, which represents enabling that level and all lower−level debug messages. The more comprehensive option is to specify a hexadecimal bitmask to enable or disable specific levels (such as if you wish to suppress a level).

Currently supported debug levels:

0, *0x0010*: Fatal failures. Anything that would prevent SSSD from starting up or causes it to cease running.

1, *0x0020*: Critical failures. An error that doesn´t kill the SSSD, but one that indicates that at least one major feature is not going to work properly.

2, *0x0040*: Serious failures. An error announcing that a particular request or operation has failed.

3, *0x0080*: Minor failures. These are the errors that would percolate down to cause the operation failure of 2.

4, *0x0100*: Configuration settings.

5, *0x0200*: Function data.

6, *0x0400*: Trace messages for operation functions.

7, *0x1000*: Trace messages for internal control functions.

8, *0x2000*: Contents of function−internal variables that may be interesting.

9, *0x4000*: Extremely low−level tracing information.

To log required bitmask debug levels, simply add their numbers together as shown in following examples:

Example: To log fatal failures, critical failures, serious failures and function data use 0x0270.

Example: To log fatal failures, configuration settings, function data, trace messages for internal control functions use 0x1310.

Note: The bitmask format of debug levels was introduced in 1.7.0.

Default: 0

debug_timestamps (bool)
Add a timestamp to the debug messages. If journald is enabled for SSSD debug logging this option is ignored.

Default: true

debug_microseconds (bool)
Add microseconds to the timestamp in debug messages. If journald is enabled for SSSD debug logging this option is ignored.

Default: false

Options usable in SERVICE and DOMAIN sections

timeout (integer)
Timeout in seconds between heartbeats for this service. This is used to ensure that the process is alive and capable of answering requests.

Default: 10

SPECIAL SECTIONS

The [sssd] section

Individual pieces of SSSD functionality are provided by special SSSD services that are started and stopped together with SSSD. The services are managed by a special service frequently called “monitor”. The “[sssd]” section is used to configure the monitor as well as some other important options like the identity domains.

Section parameters

config_file_version (integer)
Indicates what is the syntax of the config file. SSSD 0.6.0 and later use version 2.

services
Comma separated list of services that are started when sssd itself starts.

Supported services: nss, pam , sudo , autofs , ssh , pac , ifp

reconnection_retries (integer)
Number of times services should attempt to reconnect in the event of a Data Provider crash or restart before they give up

Default: 3

domains
A domain is a database containing user information. SSSD can use more domains at the same time, but at least one must be configured or SSSD won´t start. This parameter described the list of domains in the order you want them to be queried. A domain name should only consist of alphanumeric ASCII characters, dashes, dots and underscores.

re_expression (string)
Default regular expression that describes how to parse the string containing user name and domain into these components.

Each domain can have an individual regular expression configured. For some ID providers there are also default regular expressions. See DOMAIN SECTIONS for more info on these regular expressions.

full_name_format (string)
A **printf**(3)−compatible format that describes how to compose a fully qualified name from user name and domain name components.

The following expansions are supported:

%1$s
user name

%2$s
domain name as specified in the SSSD config file.

%3$s
domain flat name. Mostly usable for Active Directory domains, both directly configured or discovered via IPA trusts.

Each domain can have an individual format string configured. see DOMAIN SECTIONS for more info on this option.

try_inotify (boolean)
SSSD monitors the state of resolv.conf to identify when it needs to update its internal DNS resolver. By default, we will attempt to use inotify for this, and will fall back to polling resolv.conf every five seconds if inotify cannot be used.

There are some limited situations where it is preferred that we should skip even trying to use inotify. In these rare cases, this option should be set to ´false´

Default: true on platforms where inotify is supported. False on other platforms.

Note: this option will have no effect on platforms where inotify is unavailable. On these platforms, polling will always be used.

krb5_rcache_dir (string)
Directory on the filesystem where SSSD should store Kerberos replay cache files.

This option accepts a special value __LIBKRB5_DEFAULTS__ that will instruct SSSD to let libkrb5 decide the appropriate location for the replay cache.

Default: Distribution−specific and specified at build−time. (__LIBKRB5_DEFAULTS__ if not

configured)

user (string)
The user to drop the privileges to where appropriate to avoid running as the root user.

Default: not set, process will run as root

default_domain_suffix (string)
This string will be used as a default domain name for all names without a domain name component. The main use case is environments where the primary domain is intended for managing host policies and all users are located in a trusted domain. The option allows those users to log in just with their user name without giving a domain name as well.

Please note that if this option is set all users from the primary domain have to use their fully qualified name, e.g. user@domain.name, to log in.

Default: not set

override_space (string)
This parameter will replace spaces (space bar) with the given character for user and group names. e.g. (_). User name "john doe" will be "john_doe" This feature was added to help compatibility with shell scripts that have difficulty handling spaces, due to the default field separator in the shell.

Please note it is a configuration error to use a replacement character that might be used in user or group names. If a name contains the replacement character SSSD tries to return the unmodified name but in general the result of a lookup is undefined.

Default: not set (spaces will not be replaced)

SERVICES SECTIONS

Settings that can be used to configure different services are described in this section. They should reside in the [*$NAME*] section, for example, for NSS service, the section would be “[nss]”

General service configuration options

These options can be used to configure any service.

reconnection_retries (integer)
Number of times services should attempt to reconnect in the event of a Data Provider crash or restart before they give up

Default: 3

fd_limit
This option specifies the maximum number of file descriptors that may be opened at one time by this SSSD process. On systems where SSSD is granted the CAP_SYS_RESOURCE capability, this will be an absolute setting. On systems without this capability, the resulting value will be the lower value of this or the limits.conf "hard" limit.

Default: 8192 (or limits.conf "hard" limit)

client_idle_timeout
This option specifies the number of seconds that a client of an SSSD process can hold onto a file descriptor without communicating on it. This value is limited in order to avoid resource exhaustion on the system.

Default: 60

force_timeout (integer)
If a service is not responding to ping checks (see the “timeout” option), it is first sent the SIGTERM signal that instructs it to quit gracefully. If the service does not terminate after “force_timeout”

seconds, the monitor will forcibly shut it down by sending a SIGKILL signal.

Default: 60

offline_timeout (integer)
When SSSD switches to offline mode the amount of time before it tries to go back online will increase based upon the time spent disconnected. This value is in seconds and calculated by the following:

offline_timeout + random_offset

The random offset can increment up to 30 seconds. After each unsuccessful attempt to go online, the new interval is recalculated by the following:

new_interval = old_interval*2 + random_offset

Note that the maximum length of each interval is currently limited to one hour. If the calculated length of new_interval is greater than an hour, it will be forced to one hour.

Default: 60

subdomain_inherit (string)
Specifies a list of configuration parameters that should be inherited by a subdomain. Please note that only selected parameters can be inherited. Currently the following options can be inherited:

ignore_group_members

ldap_purge_cache_timeout

ldap_use_tokengroups

ldap_user_principal

Example:

subdomain_inherit = ldap_purge_cache_timeout

Default: none

NSS configuration options
These options can be used to configure the Name Service Switch (NSS) service.

enum_cache_timeout (integer)
How many seconds should nss_sss cache enumerations (requests for info about all users)

Default: 120

entry_cache_nowait_percentage (integer)
The entry cache can be set to automatically update entries in the background if they are requested beyond a percentage of the entry_cache_timeout value for the domain.

For example, if the domain´s entry_cache_timeout is set to 30s and entry_cache_nowait_percentage is set to 50 (percent), entries that come in after 15 seconds past the last cache update will be returned immediately, but the SSSD will go and update the cache on its own, so that future requests will not need to block waiting for a cache update.

Valid values for this option are 0−99 and represent a percentage of the entry_cache_timeout for each

domain. For performance reasons, this percentage will never reduce the nowait timeout to less than 10 seconds. (0 disables this feature)

Default: 50

entry_negative_timeout (integer)
Specifies for how many seconds nss_sss should cache negative cache hits (that is, queries for invalid database entries, like nonexistent ones) before asking the back end again.

Default: 15

filter_users, filter_groups (string)
Exclude certain users from being fetched from the sss NSS database. This is particularly useful for system accounts. This option can also be set per−domain or include fully−qualified names to filter only users from the particular domain.

Default: root

filter_users_in_groups (bool)
If you want filtered user still be group members set this option to false.

Default: true

override_homedir (string)
Override the user´s home directory. You can either provide an absolute value or a template. In the template, the following sequences are substituted:

%u
login name

%U
UID number

%d
domain name

%f
fully qualified user name (user@domain)

%P
UPN − User Principal Name (name@REALM)

%o
The original home directory retrieved from the identity provider.

%H
The value of configure option *homedir_substring*.

%%
a literal ´%´

This option can also be set per−domain.

example:

override_homedir = /home/%u

Default: Not set (SSSD will use the value retrieved from LDAP)

homedir_substring (string)
The value of this option will be used in the expansion of the *override_homedir* option if the template

contains the format string *%H*. An LDAP directory entry can directly contain this template so that this option can be used to expand the home directory path for each client machine (or operating system). It can be set per−domain or globally in the [nss] section. A value specified in a domain section will override one set in the [nss] section.

Default: /home

fallback_homedir (string)
Set a default template for a user´s home directory if one is not specified explicitly by the domain´s data provider.

The available values for this option are the same as for override_homedir.

example:

fallback_homedir = /home/%u

Default: not set (no substitution for unset home directories)

override_shell (string)
Override the login shell for all users. This option supersedes any other shell options if it takes effect and can be set either in the [nss] section or per−domain.

Default: not set (SSSD will use the value retrieved from LDAP)

allowed_shells (string)
Restrict user shell to one of the listed values. The order of evaluation is:

1. If the shell is present in “/etc/shells”, it is used.

2. If the shell is in the allowed_shells list but not in “/etc/shells”, use the value of the shell_fallback parameter.

3. If the shell is not in the allowed_shells list and not in “/etc/shells”, a nologin shell is used.

The wildcard (*) can be used to allow any shell.

The (*) is useful if you want to use shell_fallback in case that user´s shell is not in “/etc/shells” and maintaining list of all allowed shells in allowed_shells would be to much overhead.

An empty string for shell is passed as−is to libc.

The “/etc/shells” is only read on SSSD start up, which means that a restart of the SSSD is required in case a new shell is installed.

Default: Not set. The user shell is automatically used.

vetoed_shells (string)
Replace any instance of these shells with the shell_fallback

shell_fallback (string)
The default shell to use if an allowed shell is not installed on the machine.

Default: /bin/sh

default_shell
The default shell to use if the provider does not return one during lookup. This option can be specified

globally in the [nss] section or per−domain.

Default: not set (Return NULL if no shell is specified and rely on libc to substitute something sensible when necessary, usually /bin/sh)

get_domains_timeout (int)
Specifies time in seconds for which the list of subdomains will be considered valid.

Default: 60

memcache_timeout (int)
Specifies time in seconds for which records in the in−memory cache will be valid

Default: 300

user_attributes (string)
Some of the additional NSS responder requests can return more attributes than just the POSIX ones defined by the NSS interface. The list of attributes is controlled by this option. It is handle the same way as the “user_attributes” option of the InfoPipe responder (see **sssd-ifp**(5) for details) but with no default values.

To make configuration more easy the NSS responder will check the InfoPipe option if it is not set for the NSS responder.

Default: not set, fallback to InfoPipe option

PAM configuration options
These options can be used to configure the Pluggable Authentication Module (PAM) service.

offline_credentials_expiration (integer)
If the authentication provider is offline, how long should we allow cached logins (in days since the last successful online login).

Default: 0 (No limit)

offline_failed_login_attempts (integer)
If the authentication provider is offline, how many failed login attempts are allowed.

Default: 0 (No limit)

offline_failed_login_delay (integer)
The time in minutes which has to pass after offline_failed_login_attempts has been reached before a new login attempt is possible.

If set to 0 the user cannot authenticate offline if offline_failed_login_attempts has been reached. Only a successful online authentication can enable offline authentication again.

Default: 5

pam_verbosity (integer)
Controls what kind of messages are shown to the user during authentication. The higher the number to more messages are displayed.

Currently sssd supports the following values:

0: do not show any message

1: show only important messages

2: show informational messages

3: show all messages and debug information

Default: 1

pam_id_timeout (integer)
For any PAM request while SSSD is online, the SSSD will attempt to immediately update the cached identity information for the user in order to ensure that authentication takes place with the latest information.

A complete PAM conversation may perform multiple PAM requests, such as account management and session opening. This option controls (on a per−client−application basis) how long (in seconds) we can cache the identity information to avoid excessive round−trips to the identity provider.

Default: 5

pam_pwd_expiration_warning (integer)
Display a warning N days before the password expires.

Please note that the backend server has to provide information about the expiration time of the password. If this information is missing, sssd cannot display a warning.

If zero is set, then this filter is not applied, i.e. if the expiration warning was received from backend server, it will automatically be displayed.

This setting can be overridden by setting *pwd_expiration_warning* for a particular domain.

Default: 0

get_domains_timeout (int)
Specifies time in seconds for which the list of subdomains will be considered valid.

Default: 60

pam_trusted_users (string)
Specifies the comma−separated list of UID values or user names that are allowed to access the PAM responder. User names are resolved to UIDs at startup.

Default: all (All users are allowed to access the PAM responder)

Please note that UID 0 is always allowed to access the PAM responder even in case it is not in the pam_trusted_users list.

pam_public_domains (string)
Specifies the comma−separated list of domain names that are accessible even to untrusted users.

Two special values for pam_public_domains option are defined:

all (Untrusted users are allowed to access all domains in PAM responder.)

none (Untrusted users are not allowed to access any domains PAM in responder.)

Default: none

pam_account_expired_message (string)
If user is authenticating using SSH keys and account is expired then by default ´Permission denied´ is output. This output will be changed to content of this variable if it is set.

example:

pam_account_expired_message = Account expired, please call help desk.

Default: none

SUDO configuration options
These options can be used to configure the sudo service. The detailed instructions for configuration of **sudo**(8) to work with **sssd**(8) are in the manual page **sssd-sudo**(5).

sudo_timed (bool)
Whether or not to evaluate the sudoNotBefore and sudoNotAfter attributes that implement time−dependent sudoers entries.

Default: false

AUTOFS configuration options
These options can be used to configure the autofs service.

autofs_negative_timeout (integer)
Specifies for how many seconds should the autofs responder negative cache hits (that is, queries for invalid map entries, like nonexistent ones) before asking the back end again.

Default: 15

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.

SSH configuration options
These options can be used to configure the SSH service.

ssh_hash_known_hosts (bool)
Whether or not to hash host names and addresses in the managed known_hosts file.

Default: true

ssh_known_hosts_timeout (integer)
How many seconds to keep a host in the managed known_hosts file after its host keys were requested.

Default: 180

PAC responder configuration options
The PAC responder works together with the authorization data plugin for MIT Kerberos sssd_pac_plugin.so and a sub−domain provider. The plugin sends the PAC data during a GSSAPI authentication to the PAC responder. The sub−domain provider collects domain SID and ID ranges of the domain the client is joined to and of remote trusted domains from the local domain controller. If the PAC is decoded and evaluated some of the following operations are done:

- If the remote user does not exist in the cache, it is created. The uid is determined with the help of the SID, trusted domains will have UPGs and the gid will have the same value as the uid. The home directory is set based on the subdomain_homedir parameter. The shell will be empty by default, i.e. the system defaults are used, but can be overwritten with the default_shell parameter.
- If there are SIDs of groups from domains sssd knows about, the user will be added to those groups.

These options can be used to configure the PAC responder.

allowed_uids (string)
Specifies the comma−separated list of UID values or user names that are allowed to access the PAC responder. User names are resolved to UIDs at startup.

Default: 0 (only the root user is allowed to access the PAC responder)

Please note that although the UID 0 is used as the default it will be overwritten with this option. If you still want to allow the root user to access the PAC responder, which would be the typical case, you have to add 0 to the list of allowed UIDs as well.

DOMAIN SECTIONS

These configuration options can be present in a domain configuration section, that is, in a section called "[domain/*NAME*]"

min_id,max_id (integer)
UID and GID limits for the domain. If a domain contains an entry that is outside these limits, it is ignored.

For users, this affects the primary GID limit. The user will not be returned to NSS if either the UID or the primary GID is outside the range. For non−primary group memberships, those that are in range will be reported as expected.

These ID limits affect even saving entries to cache, not only returning them by name or ID.

Default: 1 for min_id, 0 (no limit) for max_id

enumerate (bool)
Determines if a domain can be enumerated. This parameter can have one of the following values:

TRUE = Users and groups are enumerated

FALSE = No enumerations for this domain

Default: FALSE

Note: Enabling enumeration has a moderate performance impact on SSSD while enumeration is running. It may take up to several minutes after SSSD startup to fully complete enumerations. During this time, individual requests for information will go directly to LDAP, though it may be slow, due to the heavy enumeration processing. Saving a large number of entries to cache after the enumeration completes might also be CPU intensive as the memberships have to be recomputed.

While the first enumeration is running, requests for the complete user or group lists may return no results until it completes.

Further, enabling enumeration may increase the time necessary to detect network disconnection, as longer timeouts are required to ensure that enumeration lookups are completed successfully. For more information, refer to the man pages for the specific id_provider in use.

For the reasons cited above, enabling enumeration is not recommended, especially in large environments.

subdomain_enumerate (string)
Whether any of autodetected trusted domains should be enumerated. The supported values are:

all
All discovered trusted domains will be enumerated

none
No discovered trusted domains will be enumerated

Optionally, a list of one or more domain names can enable enumeration just for these trusted domains.

Default: none

force_timeout (integer)
If a service is not responding to ping checks (see the “timeout” option), it is first sent the SIGTERM signal that instructs it to quit gracefully. If the service does not terminate after “force_timeout” seconds, the monitor will forcibly shut it down by sending a SIGKILL signal.

Default: 60

entry_cache_timeout (integer)
How many seconds should nss_sss consider entries valid before asking the backend again

The cache expiration timestamps are stored as attributes of individual objects in the cache. Therefore, changing the cache timeout only has effect for newly added or expired entries. You should run the **sss_cache**(8) tool in order to force refresh of entries that have already been cached.

Default: 5400

entry_cache_user_timeout (integer)
How many seconds should nss_sss consider user entries valid before asking the backend again

Default: entry_cache_timeout

entry_cache_group_timeout (integer)
How many seconds should nss_sss consider group entries valid before asking the backend again

Default: entry_cache_timeout

entry_cache_netgroup_timeout (integer)
How many seconds should nss_sss consider netgroup entries valid before asking the backend again

Default: entry_cache_timeout

entry_cache_service_timeout (integer)
How many seconds should nss_sss consider service entries valid before asking the backend again

Default: entry_cache_timeout

entry_cache_sudo_timeout (integer)
How many seconds should sudo consider rules valid before asking the backend again

Default: entry_cache_timeout

entry_cache_autofs_timeout (integer)
How many seconds should the autofs service consider automounter maps valid before asking the backend again

Default: entry_cache_timeout

entry_cache_ssh_host_timeout (integer)
How many seconds to keep a host ssh key after refresh. IE how long to cache the host key for.

Default: entry_cache_timeout

refresh_expired_interval (integer)

Specifies how many seconds SSSD has to wait before triggering a background refresh task which will refresh all expired or nearly expired records.

The background refresh will process users, groups and netgroups in the cache.

You can consider setting this value to 3/4 * entry_cache_timeout.

Default: 0 (disabled)

cache_credentials (bool)
Determines if user credentials are also cached in the local LDB cache

User credentials are stored in a SHA512 hash, not in plaintext

Default: FALSE

account_cache_expiration (integer)
Number of days entries are left in cache after last successful login before being removed during a cleanup of the cache. 0 means keep forever. The value of this parameter must be greater than or equal to offline_credentials_expiration.

Default: 0 (unlimited)

pwd_expiration_warning (integer)
Display a warning N days before the password expires.

If zero is set, then this filter is not applied, i.e. if the expiration warning was received from backend server, it will automatically be displayed.

Please note that the backend server has to provide information about the expiration time of the password. If this information is missing, sssd cannot display a warning. Also an auth provider has to be configured for the backend.

Default: 7 (Kerberos), 0 (LDAP)

id_provider (string)
The identification provider used for the domain. Supported ID providers are:

“proxy”: Support a legacy NSS provider

“local”: SSSD internal provider for local users

“ldap”: LDAP provider. See **sssd-ldap**(5) for more information on configuring LDAP.

“ipa”: FreeIPA and Red Hat Enterprise Identity Management provider. See **sssd-ipa**(5) for more information on configuring FreeIPA.

“ad”: Active Directory provider. See **sssd-ad**(5) for more information on configuring Active Directory.

use_fully_qualified_names (bool)
Use the full name and domain (as formatted by the domain´s full_name_format) as the user´s login name reported to NSS.

If set to TRUE, all requests to this domain must use fully qualified names. For example, if used in LOCAL domain that contains a "test" user, **getent passwd test** wouldn´t find the user while **getent passwd test@LOCAL** would.

NOTE: This option has no effect on netgroup lookups due to their tendency to include nested netgroups without qualified names. For netgroups, all domains will be searched when an unqualified name is requested.

Default: FALSE

ignore_group_members (bool)
Do not return group members for group lookups.

If set to TRUE, the group membership attribute is not requested from the ldap server, and group members are not returned when processing group lookup calls.

Default: FALSE

auth_provider (string)
The authentication provider used for the domain. Supported auth providers are:

“ldap” for native LDAP authentication. See **sssd-ldap**(5) for more information on configuring LDAP.

“krb5” for Kerberos authentication. See **sssd-krb5**(5) for more information on configuring Kerberos.

“ipa”: FreeIPA and Red Hat Enterprise Identity Management provider. See **sssd-ipa**(5) for more information on configuring FreeIPA.

“ad”: Active Directory provider. See **sssd-ad**(5) for more information on configuring Active Directory.

“proxy” for relaying authentication to some other PAM target.

“local”: SSSD internal provider for local users

“none” disables authentication explicitly.

Default: “id_provider” is used if it is set and can handle authentication requests.

access_provider (string)
The access control provider used for the domain. There are two built−in access providers (in addition to any included in installed backends) Internal special providers are:

“permit” always allow access. It´s the only permitted access provider for a local domain.

“deny” always deny access.

"ldap" for native LDAP authentication. See **sssd-ldap**(5) for more information on configuring LDAP.

"ipa": FreeIPA and Red Hat Enterprise Identity Management provider. See **sssd-ipa**(5) for more information on configuring FreeIPA.

"ad": Active Directory provider. See **sssd-ad**(5) for more information on configuring Active Directory.

"simple" access control based on access or deny lists. See **sssd-simple**(5) for more information on configuring the simple access module.

Default: "permit"

chpass_provider (string)
The provider which should handle change password operations for the domain. Supported change password providers are:

"ldap" to change a password stored in a LDAP server. See **sssd-ldap**(5) for more information on configuring LDAP.

"krb5" to change the Kerberos password. See **sssd-krb5**(5) for more information on configuring Kerberos.

"ipa": FreeIPA and Red Hat Enterprise Identity Management provider. See **sssd-ipa**(5) for more information on configuring FreeIPA.

"ad": Active Directory provider. See **sssd-ad**(5) for more information on configuring Active Directory.

"proxy" for relaying password changes to some other PAM target.

"none" disallows password changes explicitly.

Default: "auth_provider" is used if it is set and can handle change password requests.

sudo_provider (string)
The SUDO provider used for the domain. Supported SUDO providers are:

"ldap" for rules stored in LDAP. See **sssd-ldap**(5) for more information on configuring LDAP.

"ipa" the same as "ldap" but with IPA default settings.

"ad" the same as "ldap" but with AD default settings.

"none" disables SUDO explicitly.

Default: The value of “id_provider” is used if it is set.

The detailed instructions for configuration of sudo_provider are in the manual page **sssd-sudo**(5). There are many configuration options that can be used to adjust the behavior. Please refer to "ldap_sudo_*" in **sssd-ldap**(5).

selinux_provider (string)
The provider which should handle loading of selinux settings. Note that this provider will be called right after access provider ends. Supported selinux providers are:

“ipa” to load selinux settings from an IPA server. See **sssd-ipa**(5) for more information on configuring IPA.

“none” disallows fetching selinux settings explicitly.

Default: “id_provider” is used if it is set and can handle selinux loading requests.

subdomains_provider (string)
The provider which should handle fetching of subdomains. This value should be always the same as id_provider. Supported subdomain providers are:

“ipa” to load a list of subdomains from an IPA server. See **sssd-ipa**(5) for more information on configuring IPA.

“ad” to load a list of subdomains from an Active Directory server. See **sssd-ad**(5) for more information on configuring the AD provider.

“none” disallows fetching subdomains explicitly.

Default: The value of “id_provider” is used if it is set.

autofs_provider (string)
The autofs provider used for the domain. Supported autofs providers are:

“ldap” to load maps stored in LDAP. See **sssd-ldap**(5) for more information on configuring LDAP.

“ipa” to load maps stored in an IPA server. See **sssd-ipa**(5) for more information on configuring IPA.

“none” disables autofs explicitly.

Default: The value of “id_provider” is used if it is set.

hostid_provider (string)
The provider used for retrieving host identity information. Supported hostid providers are:

“ipa” to load host identity stored in an IPA server. See **sssd-ipa**(5) for more information on configuring IPA.

“none” disables hostid explicitly.

Default: The value of “id_provider” is used if it is set.

re_expression (string)
Regular expression for this domain that describes how to parse the string containing user name and domain into these components. The "domain" can match either the SSSD configuration domain name, or, in the case of IPA trust subdomains and Active Directory domains, the flat (NetBIOS) name of the domain.

Default for the AD and IPA provider:
“(((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?P<domain>.+$))|(^(?P<name>[^@\\]+)$))” which allows three different styles for user names:

- username
- username@domain.name
- domain\username

While the first two correspond to the general default the third one is introduced to allow easy integration of users from Windows domains.

Default: “(?P<name>[^@]+)@?(?P<domain>[^@]*$)” which translates to "the name is everything up to the “@” sign, the domain everything after that"

PLEASE NOTE: the support for non−unique named subpatterns is not available on all platforms (e.g. RHEL5 and SLES10). Only platforms with libpcre version 7 or higher can support non−unique named subpatterns.

PLEASE NOTE ALSO: older version of libpcre only support the Python syntax (?P<name>) to label subpatterns.

full_name_format (string)
A **printf**(3)−compatible format that describes how to compose a fully qualified name from user name and domain name components.

The following expansions are supported:

%1$s
user name

%2$s
domain name as specified in the SSSD config file.

%3$s
domain flat name. Mostly usable for Active Directory domains, both directly configured or discovered via IPA trusts.

Default: “%1$s@%2$s”.

lookup_family_order (string)
Provides the ability to select preferred address family to use when performing DNS lookups.

Supported values:

ipv4_first: Try looking up IPv4 address, if that fails, try IPv6

ipv4_only: Only attempt to resolve hostnames to IPv4 addresses.

ipv6_first: Try looking up IPv6 address, if that fails, try IPv4

ipv6_only: Only attempt to resolve hostnames to IPv6 addresses.

Default: ipv4_first

dns_resolver_timeout (integer)
Defines the amount of time (in seconds) to wait for a reply from the DNS resolver before assuming that it is unreachable. If this timeout is reached, the domain will continue to operate in offline mode.

Default: 6

dns_discovery_domain (string)
If service discovery is used in the back end, specifies the domain part of the service discovery DNS query.

Default: Use the domain part of machine´s hostname

override_gid (integer)
Override the primary GID value with the one specified.

case_sensitive (string)
Treat user and group names as case sensitive. At the moment, this option is not supported in the local provider. Possible option values are:

True
Case sensitive. This value is invalid for AD provider.

False
Case insensitive.

Preserving
Same as False (case insensitive), but does not lowercase names in the result of NSS operations. Note that name aliases (and in case of services also protocol names) are still lowercased in the output.

Default: True (False for AD provider)

proxy_fast_alias (boolean)
When a user or group is looked up by name in the proxy provider, a second lookup by ID is performed to "canonicalize" the name in case the requested name was an alias. Setting this option to true would cause the SSSD to perform the ID lookup from cache for performance reasons.

Default: false

subdomain_homedir (string)
Use this homedir as default value for all subdomains within this domain in IPA AD trust. See *override_homedir* for info about possible values. In addition to those, the expansion below can only be used with *subdomain_homedir*.

%F
flat (NetBIOS) name of a subdomain.

The value can be overridden by *override_homedir* option.

Default: /home/%d/%u

realmd_tags (string)
Various tags stored by the realmd configuration service for this domain.

Options valid for proxy domains.

proxy_pam_target (string)

The proxy target PAM proxies to.

Default: not set by default, you have to take an existing pam configuration or create a new one and add the service name here.

proxy_lib_name (string)
The name of the NSS library to use in proxy domains. The NSS functions searched for in the library are in the form of _nss_$(libName)_$(function), for example _nss_files_getpwent.

The local domain section

This section contains settings for domain that stores users and groups in SSSD native database, that is, a domain that uses *id_provider=local*.

Section parameters

default_shell (string)
The default shell for users created with SSSD userspace tools.

Default: /bin/bash

base_directory (string)
The tools append the login name to *base_directory* and use that as the home directory.

Default: /home

create_homedir (bool)
Indicate if a home directory should be created by default for new users. Can be overridden on command line.

Default: TRUE

remove_homedir (bool)
Indicate if a home directory should be removed by default for deleted users. Can be overridden on command line.

Default: TRUE

homedir_umask (integer)
Used by **sss_useradd**(8) to specify the default permissions on a newly created home directory.

Default: 077

skel_dir (string)
The skeleton directory, which contains files and directories to be copied in the user´s home directory, when the home directory is created by **sss_useradd**(8)

Default: /etc/skel

mail_dir (string)
The mail spool directory. This is needed to manipulate the mailbox when its corresponding user account is modified or deleted. If not specified, a default value is used.

Default: /var/mail

userdel_cmd (string)
The command that is run after a user is removed. The command us passed the username of the user being removed as the first and only parameter. The return code of the command is not taken into account.

Default: None, no command is run

EXAMPLE

The following example shows a typical SSSD config. It does not describe configuration of the domains themselves − refer to documentation on configuring domains for more details.

```
[sssd]
domains = LDAP
services = nss, pam
config_file_version = 2

[nss]
filter_groups = root
filter_users = root

[pam]

[domain/LDAP]
id_provider = ldap
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com

auth_provider = krb5
krb5_server = kerberos.example.com
krb5_realm = EXAMPLE.COM
cache_credentials = true

min_id = 10000
max_id = 20000
enumerate = False
```

SEE ALSO

sssd(8), **sssd.conf**(5), **sssd-ldap**(5), **sssd-krb5**(5), **sssd-simple**(5), **sssd-ipa**(5), **sssd-ad**(5), **sssd-sudo**(5), **sss_cache**(8), **sss_debuglevel**(8), **sss_groupadd**(8), **sss_groupdel**(8), **sss_groupshow**(8), **sss_groupmod**(8), **sss_useradd**(8), **sss_userdel**(8), **sss_usermod**(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)

AUTHORS

The SSSD upstream − http://fedorahosted.org/sssd