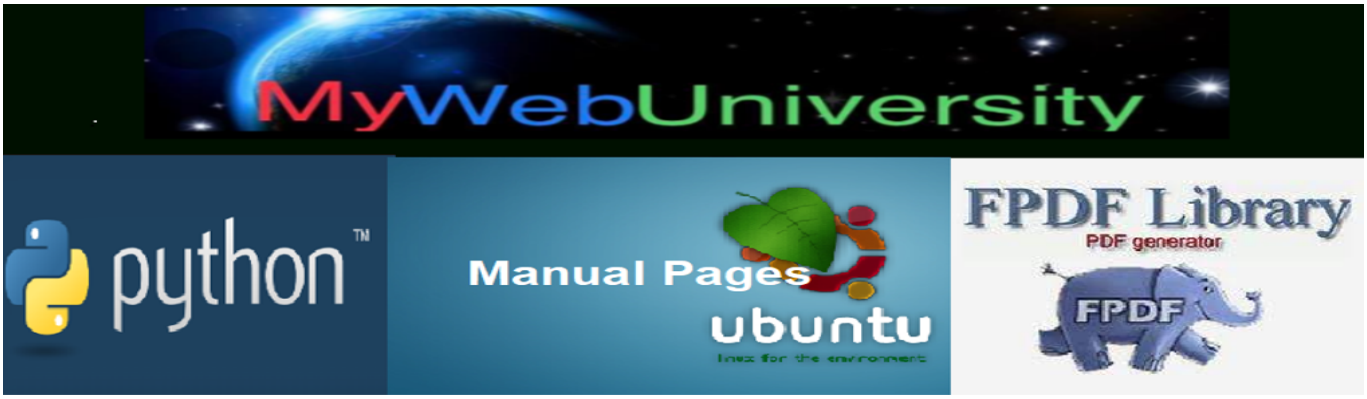

Rocky Enterprise Linux 9.2 Manual Pages on command 'sssd.conf.5'

$ man sssd.conf.5

SSSD.CONF(5) File Formats and Conventions SSSD.CONF(5)

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 comment line 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.

CONFIGURATION SNIPPETS FROM INCLUDE DIRECTORY

The configuration file sssd.conf will include configuration snippets using the include directory conf.d. This feature is available if SSSD was compiled with libini version 1.3.0 or later.

Any file placed in conf.d that ends in ?.conf? and does not begin with a dot (?.?) will be

used together with sssd.conf to configure SSSD.

The configuration snippets from conf.d have higher priority than sssd.conf and will override sssd.conf when conflicts occur. If several snippets are present in conf.d, then they are included in alphabetical order (based on locale). Files included later have higher priority. Numerical prefixes (01_snippet.conf, 02_snippet.conf etc.) can help visualize the priority (higher number means higher priority).

The snippet files require the same owner and permissions as sssd.conf. Which are by default root:root and 0600.

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).

Please note that each SSSD service logs into its own log file. Also please note that enabling ?debug_level? in the ?[sssd]? section only enables debugging just for the sssd process itself, not for the responder or provider processes. The ?debug_level? parameter should be added to all sections that you wish to produce debug logs from.

In addition to changing the log level in the config file using the ?debug_level? parameter, which is persistent, but requires SSSD restart, it is also possible to change the debug level on the fly using the sss_debuglevel(8) tool.

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 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.

10, 0x10000: Even more low-level libldb tracing information. Almost never really required.

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: 0x0070 (i.e. fatal, critical and serious failures; corresponds to setting 2 in decimal notation)

debug (integer)

SSSD 1.14 and later also includes the debug alias for debug_level as a convenience feature. If both are specified, the value of debug_level will be used.

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

debug_backtrace_enabled (bool)

Enable debug backtrace.

In case SSSD is run with debug_level less than 9, everything is logged to a ring buffer in memory and flushed to a log file on any error up to and including `min(0x0040, debug_level)` (i.e. if debug_level is explicitly set to 0 or 1 then only

those error levels will trigger backtrace, otherwise up to 2).

Feature is only supported for `logger == files` (i.e. setting doesn't have effect for other logger types).

Default: true

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. Note that after three missed heartbeats the process will terminate itself.

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. The services' list is optional on platforms where systemd is supported, as they will either be socket or D-Bus activated when needed.

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

By default, all services are disabled and the administrator must enable the ones allowed to be used by executing: "systemctl enable sssd-@service@.socket".

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

describes the list of domains in the order you want them to be queried. A domain name is recommended to contain only alphanumeric ASCII characters, dashes, dots and underscores. '/' character is forbidden.

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.

monitor_resolv_conf (boolean)

Controls if SSSD should monitor the state of resolv.conf to identify when it needs to update its internal DNS resolver.

Default: true

try_inotify (boolean)

By default, SSSD will attempt to use inotify to monitor configuration files changes and will fall back to polling 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. This option does not work when running socket-activated services, as the user set up to run the processes is set up during compilation time. The way to override the systemd unit files is by creating the appropriate files in /etc/systemd/system/. Keep in mind that any change in the socket user, group or permissions may result in a non-usable SSSD. The same may occur in case of changes of the user running the NSS responder.

Both a user name and a uid can be used but the user should be a local one, i.e. accessible via ?files? service of nsswitch.conf.

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. Setting this option changes default of use_fully_qualified_names to True. It is not allowed to use this option together with use_fully_qualified_names set to False. One exception from this rule are domains with ?id_provider=files? that always try to match the behaviour of nss_files and therefore their output is not qualified even when the default_domain_suffix option is used.

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)

certificate_verification (string)

With this parameter the certificate verification can be tuned with a comma separated list of options. Supported options are:

no_ocsp

Disables Online Certificate Status Protocol (OCSP) checks. This might be needed if the OCSP servers defined in the certificate are not reachable from the client.

soft_ocsp

If a connection cannot be established to an OCSP responder the OCSP check is skipped. This option should be used to allow authentication when the system is offline and the OCSP responder cannot be reached.

ocsp_dgst

Digest (hash) function used to create the certificate ID for the OCSP request.

Allowed values are:

? sha1

? sha256

? sha384

? sha512

Default: sha1 (to allow compatibility with RFC5019-compliant responder)

no_verification

Disables verification completely. This option should only be used for testing.

partial_chain

Allow verification to succeed even if a complete chain cannot be built to a self-signed trust-anchor, provided it is possible to construct a chain to a trusted certificate that might not be self-signed.

ocsp_default_responder=URL

Sets the OCSP default responder which should be used instead of the one mentioned in the certificate. URL must be replaced with the URL of the OCSP default responder e.g. http://example.com:80/ocsp.

ocsp_default_responder_signing_cert=NAME

This option is currently ignored. All needed certificates must be available in the PEM file given by pam_cert_db_path.

crl_file=/PATH/TO/CRL/FILE

Use the Certificate Revocation List (CRL) from the given file during the verification of the certificate. The CRL must be given in PEM format, see crl(1ssl) for details.

soft_crl

If a Certificate Revocation List (CRL) is expired ignore the CRL checks for the related certificates. This option should be used to allow authentication when the system is offline and the CRL cannot be renewed.

Unknown options are reported but ignored.

Default: not set, i.e. do not restrict certificate verification

disable_netlink (boolean)

SSSD hooks into the netlink interface to monitor changes to routes, addresses, links and trigger certain actions.

The SSSD state changes caused by netlink events may be undesirable and can be disabled by setting this option to 'true'

Default: false (netlink changes are detected)

enable_files_domain (boolean)

When this option is enabled, SSSD prepends an implicit domain with ?id_provider=files? before any explicitly configured domains.

Default: false

domain_resolution_order

Comma separated list of domains and subdomains representing the lookup order that will be followed. The list doesn't have to include all possible domains as the missing domains will be looked up based on the order they're presented in the ?domains? configuration option. The subdomains which are not listed as part of ?lookup_order? will be looked up in a random order for each parent domain.

Please, note that when this option is set the output format of all commands is always

fully-qualified even when using short names for input, for all users but the ones managed by the files provider. In case the administrator wants the output not fully-qualified, the full_name_format option can be used as shown below: ?full_name_format=%1$s? However, keep in mind that during login, login applications often canonicalize the username by calling getpwnam(3) which, if a shortname is returned for a qualified input (while trying to reach a user which exists in multiple domains) might re-route the login attempt into the domain which uses shortnames, making this workaround totally not recommended in cases where usernames may overlap between domains.

Default: Not set

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. The timeout can't be shorter than 10 seconds. If a lower value is configured, it will be adjusted to 10 seconds.

Default: 60, KCM: 300

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. By default SSSD uses incremental behaviour to calculate delay in between retries. So, the wait time for a given retry will be longer than the wait time for the previous ones. After each unsuccessful attempt to go online, the new interval is recalculated by the following:

new_delay = Minimum(old_delay * 2, offline_timeout_max) + random[0...offline_timeout_random_offset]

The offline_timeout default value is 60. The offline_timeout_max default value is 3600. The offline_timeout_random_offset default value is 30. The end result is amount of seconds before next retry.

Note that the maximum length of each interval is defined by offline_timeout_max (apart of random part).

Default: 60

offline_timeout_max (integer)

Controls by how much the time between attempts to go online can be incremented following unsuccessful attempts to go online.

A value of 0 disables the incrementing behaviour.

The value of this parameter should be set in correlation to offline_timeout parameter value.

With offline_timeout set to 60 (default value) there is no point in setting offlinet_timeout_max to less than 120 as it will saturate instantly. General rule here should be to set offline_timeout_max to at least 4 times offline_timeout.

Although a value between 0 and offline_timeout may be specified, it has the effect of overriding the offline_timeout value so is of little use.

Default: 3600

offline_timeout_random_offset (integer)

When SSSD is in offline mode it keeps probing backend servers in specified time intervals:

new_delay = Minimum(old_delay * 2, offline_timeout_max) + random[0...offline_timeout_random_offset]

This parameter controls the value of the random offset used for the above equation.

Final random_offset value will be random number in range:

[0 - offline_timeout_random_offset]

A value of 0 disables the random offset addition.

Default: 30

responder_idle_timeout

This option specifies the number of seconds that an SSSD responder process can be up without being used. This value is limited in order to avoid resource exhaustion on the system. The minimum acceptable value for this option is 60 seconds. Setting this option to 0 (zero) means that no timeout will be set up to the responder. This option only has effect when SSSD is built with systemd support and when services are either socket or D-Bus activated.

Default: 300

cache_first

This option specifies whether the responder should query all caches before querying the Data Providers.

Default: false

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

local_negative_timeout (integer)

Specifies for how many seconds nss_sss should keep local users and groups in negative cache before trying to look it up in the back end again. Setting the option to 0 disables this feature.

Default: 14400 (4 hours)

filter_users, filter_groups (string)

Exclude certain users or groups 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 or by a user principal name (UPN).

NOTE: The filter_groups option doesn't affect inheritance of nested group members, since filtering happens after they are propagated for returning via NSS. E.g. a group having a member group filtered out will still have the member users of the latter listed.

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)

%l

The first letter of the login name.

%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)

Please note, the home directory from a specific override for the user, either locally (see sss_override(8)) or centrally managed IPA id-overrides, has a higher precedence and will be used instead of the value given by override_homedir.

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 (integer)

Specifies time in seconds for which records in the in-memory cache will be valid.

Setting this option to zero will disable the in-memory cache.

Default: 300

WARNING: Disabling the in-memory cache will have significant negative impact on SSSD's performance and should only be used for testing.

NOTE: If the environment variable SSS_NSS_USE_MEMCACHE is set to "NO", client applications will not use the fast in-memory cache.

memcache_size_passwd (integer)

Size (in megabytes) of the data table allocated inside fast in-memory cache for passwd requests. Setting the size to 0 will disable the passwd in-memory cache.

Default: 8

WARNING: Disabled or too small in-memory cache can have significant negative impact on SSSD's performance.

NOTE: If the environment variable SSS_NSS_USE_MEMCACHE is set to "NO", client applications will not use the fast in-memory cache.

memcache_size_group (integer)

Size (in megabytes) of the data table allocated inside fast in-memory cache for group requests. Setting the size to 0 will disable the group in-memory cache.

Default: 6

WARNING: Disabled or too small in-memory cache can have significant negative impact on SSSD's performance.

NOTE: If the environment variable SSS_NSS_USE_MEMCACHE is set to "NO", client applications will not use the fast in-memory cache.

memcache_size_initgroups (integer)

Size (in megabytes) of the data table allocated inside fast in-memory cache for initgroups requests. Setting the size to 0 will disable the initgroups in-memory cache.

Default: 10

WARNING: Disabled or too small in-memory cache can have significant negative impact on SSSD's performance.

NOTE: If the environment variable SSS_NSS_USE_MEMCACHE is set to "NO", client applications will not use the fast in-memory cache.

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 handled 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

pwfield (string)

The value that NSS operations that return users or groups will return for the ?password? field.

Default: ?*?

Note: This option can also be set per-domain which overwrites the value in [nss] section.

Default: ?not set? (remote domains), ?x? (the files domain), ?x? (proxy domain with nss_files and sssd-shadowutils target)

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_response_filter (string)

A comma separated list of strings which allows to remove (filter) data sent by the PAM responder to pam_sss PAM module. There are different kind of responses sent to pam_sss e.g. messages displayed to the user or environment variables which should be set by pam_sss.

While messages already can be controlled with the help of the pam_verbosity option this option allows to filter out other kind of responses as well.

Currently the following filters are supported:

ENV

Do not send any environment variables to any service.

ENV:var_name

Do not send environment variable var_name to any service.

ENV:var_name:service

Do not send environment variable var_name to service.

The list of strings can either be the list of filters which would set this list of filters and overwrite the defaults. Or each element of the list can be prefixed by a '+' or '-' character which would add the filter to the existing default or remove it from the defaults, respectively. Please note that either all list elements must have a '+' or '-' prefix or none. It is considered as an error to mix both styles.

Default: ENV:KRB5CCNAME:sudo, ENV:KRB5CCNAME:sudo-i

Example: -ENV:KRB5CCNAME:sudo-i will remove the filter from the default list

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 run PAM conversations against trusted domains. Users not included in this list can only access domains marked as public with ?pam_public_domains?. User names are resolved to UIDs at startup.

Default: All users are considered trusted by default

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)

Allows a custom expiration message to be set, replacing the default 'Permission denied' message.

Note: Please be aware that message is only printed for the SSH service unless

pam_verbosity is set to 3 (show all messages and debug information).

example:

pam_account_expired_message = Account expired, please contact help desk.

Default: none

pam_account_locked_message (string)

Allows a custom lockout message to be set, replacing the default 'Permission denied' message.

example:

pam_account_locked_message = Account locked, please contact help desk.

Default: none

pam_cert_auth (bool)

Enable certificate based Smartcard authentication. Since this requires additional communication with the Smartcard which will delay the authentication process this option is disabled by default.

Default: False

pam_cert_db_path (string)

The path to the certificate database.

Default:

? /etc/sssd/pki/sssd_auth_ca_db.pem (path to a file with trusted CA certificates in PEM format)

pam_cert_verification (string)

With this parameter the PAM certificate verification can be tuned with a comma separated list of options that override the ?certificate_verification? value in ?[sssd]? section. Supported options are the same of ?certificate_verification?.

example:

pam_cert_verification = partial_chain

Default: not set, i.e. use default ?certificate_verification? option defined in ?[sssd]? section.

p11_child_timeout (integer)

How many seconds will pam_sss wait for p11_child to finish.

Default: 10

pam_app_services (string)

Which PAM services are permitted to contact domains of type ?application?

Default: Not set

pam_p11_allowed_services (integer)

A comma-separated list of PAM service names for which it will be allowed to use Smartcards.

It is possible to add another PAM service name to the default set by using ?+service_name? or to explicitly remove a PAM service name from the default set by using ?-service_name?. For example, in order to replace a default PAM service name for authentication with Smartcards (e.g. ?login?) with a custom PAM service name (e.g. ?my_pam_service?), you would use the following configuration:

pam_p11_allowed_services = +my_pam_service, -login

Default: the default set of PAM service names includes:

? login

? su

? su-l

? gdm-smartcard

? gdm-password

? kdm

? sudo

? sudo-i

? gnome-screensaver

p11_wait_for_card_timeout (integer)

If Smartcard authentication is required how many extra seconds in addition to p11_child_timeout should the PAM responder wait until a Smartcard is inserted.

Default: 60

p11_uri (string)

PKCS#11 URI (see RFC-7512 for details) which can be used to restrict the selection of devices used for Smartcard authentication. By default SSSD's p11_child will search for a PKCS#11 slot (reader) where the 'removable' flags is set and read the certificates from the inserted token from the first slot found. If multiple readers are connected p11_uri can be used to tell p11_child to use a specific reader.

Example:

p11_uri = pkcs11:slot-description=My%20Smartcard%20Reader

or

p11_uri = pkcs11:library-description=OpenSC%20smartcard%20framework;slot-id=2

To find suitable URI please check the debug output of p11_child. As an alternative the GnuTLS utility 'p11tool' with e.g. the '--list-all' will show PKCS#11 URIs as well.

Default: none

pam_initgroups_scheme

The PAM responder can force an online lookup to get the current group memberships of the user trying to log in. This option controls when this should be done and the following values are allowed:

always

Always do an online lookup, please note that pam_id_timeout still applies

no_session

Only do an online lookup if there is no active session of the user, i.e. if the user is currently not logged in

never

Never force an online lookup, use the data from the cache as long as they are not expired

Default: no_session

pam_gssapi_services

Comma separated list of PAM services that are allowed to try GSSAPI authentication using pam_sss_gss.so module.

To disable GSSAPI authentication, set this option to ?-? (dash).

Note: This option can also be set per-domain which overwrites the value in [pam] section. It can also be set for trusted domain which overwrites the value in the domain section.

Example:

pam_gssapi_services = sudo, sudo-i

Default: - (GSSAPI authentication is disabled)

pam_gssapi_check_upn

If True, SSSD will require that the Kerberos user principal that successfully authenticated through GSSAPI can be associated with the user who is being authenticated. Authentication will fail if the check fails.

If False, every user that is able to obtained required service ticket will be authenticated.

Note: This option can also be set per-domain which overwrites the value in [pam] section. It can also be set for trusted domain which overwrites the value in the domain section.

Default: True

pam_gssapi_indicators_map

Comma separated list of authentication indicators required to be present in a Kerberos ticket to access a PAM service that is allowed to try GSSAPI authentication using pam_sss_gss.so module.

Each element of the list can be either an authentication indicator name or a pair ?service:indicator?. Indicators not prefixed with the PAM service name will be required to access any PAM service configured to be used with pam_gssapi_services. A resulting list of indicators per PAM service is then checked against indicators in the Kerberos ticket during authentication by pam_sss_gss.so. Any indicator from the ticket that matches the resulting list of indicators for the PAM service would grant access. If none of the indicators in the list match, access will be denied. If the resulting list of indicators for the PAM service is empty, the check will not prevent the access.

To disable GSSAPI authentication indicator check, set this option to ?-? (dash). To disable the check for a specific PAM service, add ?service:-?.

Note: This option can also be set per-domain which overwrites the value in [pam] section. It can also be set for trusted domain which overwrites the value in the domain section.

Following authentication indicators are supported by IPA Kerberos deployments:

? pkinit -- pre-authentication using X.509 certificates -- whether stored in files or on smart cards.

? hardened -- SPAKE pre-authentication or any pre-authentication wrapped in a FAST channel.

? radius -- pre-authentication with the help of a RADIUS server.

? otp -- pre-authentication using integrated two-factor authentication (2FA or one-time password, OTP) in IPA.

Example: to require access to SUDO services only for users which obtained their Kerberos tickets with a X.509 certificate pre-authentication (PKINIT), set

pam_gssapi_indicators_map = sudo:pkinit, sudo-i:pkinit

Default: not set (use of authentication indicators is not required)

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

sudo_threshold (integer)

Maximum number of expired rules that can be refreshed at once. If number of expired rules is below threshold, those rules are refreshed with ?rules refresh? mechanism. If the threshold is exceeded a ?full refresh? of sudo rules is triggered instead. This threshold number also applies to IPA sudo command and command group searches.

Default: 50

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: false

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

ssh_use_certificate_keys (bool)

If set to true the sss_ssh_authorizedkeys will return ssh keys derived from the public key of X.509 certificates stored in the user entry as well. See sss_ssh_authorizedkeys(1) for details.

Default: true

ssh_use_certificate_matching_rules (string)

By default the ssh responder will use all available certificate matching rules to filter the certificates so that ssh keys are only derived from the matching ones. With this option the used rules can be restricted with a comma separated list of mapping and matching rule names. All other rules will be ignored.

There are two special key words 'all_rules' and 'no_rules' which will enable all or no rules, respectively. The latter means that no certificates will be filtered out and ssh keys will be generated from all valid certificates.

If no rules are configured using 'all_rules' will enable a default rule which enables all certificates suitable for client authentication. This is the same behavior as for the PAM responder if certificate authentication is enabled.

A non-existing rule name is considered an error. If as a result no rule is selected all certificates will be ignored.

Default: not set, equivalent to 'all_rules', all found rules or the default rule are used

ca_db (string)

Path to a storage of trusted CA certificates. The option is used to validate user certificates before deriving public ssh keys from them.

Default:

? /etc/sssd/pki/sssd_auth_ca_db.pem (path to a file with trusted CA certificates in PEM format)

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.

Local user names are required, i.e. accessible via ?files? service of nsswitch.conf.

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.

pac_lifetime (integer)

Lifetime of the PAC entry in seconds. As long as the PAC is valid the PAC data can be used to determine the group memberships of a user.

Default: 300

Session recording configuration options

Session recording works in conjunction with tlog-rec-session(8), a part of tlog package, to log what users see and type when they log in on a text terminal. See also sssd-session-recording(5).

These options can be used to configure session recording.

scope (string)

One of the following strings specifying the scope of session recording:

"none"

No users are recorded.

"some"

Users/groups specified by users and groups options are recorded.

"all"

All users are recorded.

Default: "none"

users (string)

A comma-separated list of users which should have session recording enabled. Matches user names as returned by NSS. I.e. after the possible space replacement, case changes, etc.

Default: Empty. Matches no users.

groups (string)

A comma-separated list of groups, members of which should have session recording enabled. Matches group names as returned by NSS. I.e. after the possible space replacement, case changes, etc.

NOTE: using this option (having it set to anything) has a considerable performance cost, because each uncached request for a user requires retrieving and matching the groups the user is member of.

Default: Empty. Matches no groups.

exclude_users (string)

A comma-separated list of users to be excluded from recording, only applicable with 'scope=all'.

Default: Empty. No users excluded.

exclude_groups (string)

A comma-separated list of groups, members of which should be excluded from recording. Only applicable with 'scope=all'.

NOTE: using this option (having it set to anything) has a considerable performance cost, because each uncached request for a user requires retrieving and matching the groups the user is member of.

Default: Empty. No groups excluded.

DOMAIN SECTIONS

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

enabled

Explicitly enable or disable the domain. If ?true?, the domain is always ?enabled?. If ?false?, the domain is always ?disabled?. If this option is not set, the domain is enabled only if it is listed in the domains option in the ?[sssd]? section.

domain_type (string)

Specifies whether the domain is meant to be used by POSIX-aware clients such as the Name Service Switch or by applications that do not need POSIX data to be present or generated. Only objects from POSIX domains are available to the operating system interfaces and utilities.

Allowed values for this option are ?posix? and ?application?.

POSIX domains are reachable by all services. Application domains are only reachable from the InfoPipe responder (see sssd-ifp(5)) and the PAM responder.

NOTE: The application domains are currently well tested with ?id_provider=ldap? only.

For an easy way to configure a non-POSIX domains, please see the ?Application domains? section.

Default: posix

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, that is, whether the domain can list all the users and group it contains. Note that it is not required to enable enumeration in order for secondary groups to be displayed. This parameter can have one of the following values:

TRUE = Users and groups are enumerated

FALSE = No enumerations for this domain

Default: FALSE

Enumerating a domain requires SSSD to download and store ALL user and group entries from the remote server.

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. This can lead to the ?sssd_be? process becoming unresponsive or even restarted by the internal watchdog.

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

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_resolver_timeout (integer)

How many seconds should nss_sss consider hosts and networks 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

entry_cache_computer_timeout (integer)

How many seconds to keep the local computer entry before asking the backend again

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. For users who have performed the initgroups (get group membership for user, typically ran at login) operation in the past, both the user entry and the group membership are updated.

This option is automatically inherited for all trusted domains.

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

Cache entry will be refreshed by background task when 2/3 of cache timeout has already passed. If there are existing cached entries, the background task will refer to their original cache timeout values instead of current configuration value. This may lead to a situation in which background refresh task appears to not be working. This is done by design to improve offline mode operation and reuse of existing valid cache entries.

To make this change instant the user may want to manually invalidate existing cache.

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

cache_credentials_minimal_first_factor_length (int)

If 2-Factor-Authentication (2FA) is used and credentials should be saved this value determines the minimal length the first authentication factor (long term password) must have to be saved as SHA512 hash into the cache.

This should avoid that the short PINs of a PIN based 2FA scheme are saved in the cache which would make them easy targets for brute-force attacks.

Default: 8

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.

?files?: FILES provider. See sssd-files(5) for more information on how to mirror local users and groups into SSSD.

?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 (TRUE for trusted domain/sub-domains or if default_domain_suffix is used)

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, such as getgrnam(3) or getgrgid(3). As an effect, ?getent group $groupname? would return the requested group as if it was empty.

Enabling this option can also make access provider checks for group membership

significantly faster, especially for groups containing many members.

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.

?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.

?krb5?: .k5login based access control. See sssd-krb5(5) for more information on configuring Kerberos.

?proxy? for relaying access control to another PAM 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).

NOTE: Sudo rules are periodically downloaded in the background unless the sudo provider is explicitly disabled. Set sudo_provider = None to disable all sudo-related activity in SSSD if you do not want to use sudo with SSSD at all.

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.

session_provider (string)

The provider which configures and manages user session related tasks. The only user session task currently provided is the integration with Fleet Commander, which works only with IPA. Supported session providers are:

?ipa? to allow performing user session related tasks.

?none? does not perform any kind of user session related tasks.

Default: ?id_provider? is used if it is set and can perform session related tasks.

NOTE: In order to have this feature working as expected SSSD must be running as "root" and not as the unprivileged user.

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.

?ad? to load maps stored in an AD server. See sssd-ad(5) for more information on configuring the AD provider.

?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.

resolver_provider (string)

The provider which should handle hosts and networks lookups. Supported resolver providers are:

?proxy? to forward lookups to another NSS library. See ?proxy_resolver_lib_name?

?ldap? to fetch hosts and networks stored in LDAP. See sssd-ldap(5) for more information on configuring LDAP.

?ad? to fetch hosts and networks stored in AD. See sssd-ad(5) for more information on configuring the AD provider.

?none? disallows fetching hosts and networks 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"

NOTE: Some Active Directory groups, typically those used for MS Exchange contain an ?@? sign in the name, which clashes with the default re_expression value for the AD and IPA providers. To support these groups, consider changing the re_expression value

to: ?((?P<name>.+)@(?P<domain>[^@]+$))?.

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_server_timeout (integer)

Defines the amount of time (in milliseconds) SSSD would try to talk to DNS server before trying next DNS server.

The AD provider will use this option for the CLDAP ping timeouts as well.

Please see the section ?FAILOVER? for more information about the service resolution.

Default: 1000

dns_resolver_op_timeout (integer)

Defines the amount of time (in seconds) to wait to resolve single DNS query (e.g. resolution of a hostname or an SRV record) before try next hostname or DNS discovery.

Please see the section ?FAILOVER? for more information about the service resolution.

Default: 3

dns_resolver_server_timeout (integer)

Defines the amount of time (in milliseconds) SSSD would try to talk to DNS server before trying next DNS server.

Please see the section ?FAILOVER? for more information about the service resolution.

Default: 1000

dns_resolver_op_timeout (integer)

Defines the amount of time (in seconds) to wait to resolve single DNS query (e.g. resolution of a hostname or an SRV record) before try next hostname or DNS discovery.

Please see the section ?FAILOVER? for more information about the service resolution.

Default: 3

dns_resolver_timeout (integer)

Defines the amount of time (in seconds) to wait for a reply from the internal fail over service before assuming that the service is unreachable. If this timeout is reached, the domain will continue to operate in offline mode.

Please see the section ?FAILOVER? for more information about the service resolution.

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. 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.

If you want to set this value for trusted domain with IPA provider, you need to

set it on both the client and SSSD on the server.

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

Default: True (False for AD provider)

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

ldap_krb5_keytab (the value of krb5_keytab will be used if ldap_krb5_keytab is not set

explicitly)

auto_private_groups

case_sensitive

Example:

subdomain_inherit = ldap_purge_cache_timeout

Default: none

Note: This option only works with the IPA and AD provider.

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.

cached_auth_timeout (int)

Specifies time in seconds since last successful online authentication for which user

will be authenticated using cached credentials while SSSD is in the online mode. If

the credentials are incorrect, SSSD falls back to online authentication.

This option's value is inherited by all trusted domains. At the moment it is not possible to set a different value per trusted domain.

Special value 0 implies that this feature is disabled.

Please note that if ?cached_auth_timeout? is longer than ?pam_id_timeout? then the back end could be called to handle ?initgroups.?

Default: 0

auto_private_groups (string)

This option takes any of three available values:

true

Create user's private group unconditionally from user's UID number. The GID number is ignored in this case.

NOTE: Because the GID number and the user private group are inferred from the UID number, it is not supported to have multiple entries with the same UID or GID number with this option. In other words, enabling this option enforces uniqueness across the ID space.

false

Always use the user's primary GID number. The GID number must refer to a group object in the LDAP database.

hybrid

A primary group is autogenerated for user entries whose UID and GID numbers have the same value and at the same time the GID number does not correspond to a real group object in LDAP. If the values are the same, but the primary GID in the user entry is also used by a group object, the primary GID of the user resolves to that group object.

If the UID and GID of a user are different, then the GID must correspond to a group entry, otherwise the GID is simply not resolvable.

This feature is useful for environments that wish to stop maintaining a separate group objects for the user private groups, but also wish to retain the existing user private groups.

For subdomains, the default value is False for subdomains that use assigned POSIX IDs and True for subdomains that use automatic ID-mapping.

The value of auto_private_groups can either be set per subdomains in a subsection, for example:

[domain/forest.domain/sub.domain]

auto_private_groups = false

or globally for all subdomains in the main domain section using the subdomain_inherit option:

[domain/forest.domain]

subdomain_inherit = auto_private_groups

auto_private_groups = false

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.

proxy_resolver_lib_name (string)

The name of the NSS library to use for hosts and networks lookups in proxy domains. The NSS functions searched for in the library are in the form of _nss_$(libName)_$(function), for example _nss_dns_gethostbyname2_r.

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

proxy_max_children (integer)

This option specifies the number of pre-forked proxy children. It is useful for high-load SSSD environments where sssd may run out of available child slots, which would cause some issues due to the requests being queued.

Default: 10

Application domains

SSSD, with its D-Bus interface (see sssd-ifp(5)) is appealing to applications as a gateway

to an LDAP directory where users and groups are stored. However, contrary to the traditional SSSD deployment where all users and groups either have POSIX attributes or those attributes can be inferred from the Windows SIDs, in many cases the users and groups in the application support scenario have no POSIX attributes. Instead of setting a ?[domain/NAME]? section, the administrator can set up an ?[application/NAME]? section that internally represents a domain with type ?application? optionally inherits settings from a tradition SSSD domain.

Please note that the application domain must still be explicitly enabled in the ?domains? parameter so that the lookup order between the application domain and its POSIX sibling domain is set correctly.

Application domain parameters

inherit_from (string)

The SSSD POSIX-type domain the application domain inherits all settings from. The application domain can moreover add its own settings to the application settings that augment or override the ?sibling? domain settings.

Default: Not set

The following example illustrates the use of an application domain. In this setup, the POSIX domain is connected to an LDAP server and is used by the OS through the NSS responder. In addition, the application domain also requests the telephoneNumber attribute, stores it as the phone attribute in the cache and makes the phone attribute reachable through the D-Bus interface.

```
[sssd]
domains = appdom, posixdom
[ifp]
user_attributes = +phone
[domain/posixdom]
id_provider = ldap
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
[application/appdom]
inherit_from = posixdom
ldap_user_extra_attrs = phone:telephoneNumber
```

Some options used in the domain section can also be used in the trusted domain section, that is, in a section called ?[domain/DOMAIN_NAME/TRUSTED_DOMAIN_NAME]?. Where DOMAIN_NAME is the actual joined-to base domain. Please refer to examples below for explanation.

Currently supported options in the trusted domain section are:

ldap_search_base,

ldap_user_search_base,

ldap_group_search_base,

ldap_netgroup_search_base,

ldap_service_search_base,

ldap_sasl_mech,

ad_server,

ad_backup_server,

ad_site,

use_fully_qualified_names

pam_gssapi_services

pam_gssapi_check_upn

For more details about these options see their individual description in the manual page.

CERTIFICATE MAPPING SECTION

To allow authentication with Smartcards and certificates SSSD must be able to map certificates to users. This can be done by adding the full certificate to the LDAP object of the user or to a local override. While using the full certificate is required to use the Smartcard authentication feature of SSH (see sss_ssh_authorizedkeys(8) for details) it might be cumbersome or not even possible to do this for the general case where local services use PAM for authentication.

To make the mapping more flexible mapping and matching rules were added to SSSD (see sss-certmap(5) for details).

A mapping and matching rule can be added to the SSSD configuration in a section on its own with a name like ?[certmap/DOMAIN_NAME/RULE_NAME]?. In this section the following options are allowed:

matchrule (string)

Only certificates from the Smartcard which matches this rule will be processed, all others are ignored.

Default: KRB5:<EKU>clientAuth, i.e. only certificates which have the Extended Key

Usage ?clientAuth?

maprule (string)

Defines how the user is found for a given certificate.

Default:

? LDAP:(userCertificate;binary={cert!bin}) for LDAP based providers like ?ldap?, ?AD? or ?ipa?.

? The RULE_NAME for the ?files? provider which tries to find a user with the same name.

domains (string)

Comma separated list of domain names the rule should be applied. By default a rule is only valid in the domain configured in sssd.conf. If the provider supports subdomains this option can be used to add the rule to subdomains as well.

Default: the configured domain in sssd.conf

priority (integer)

Unsigned integer value defining the priority of the rule. The higher the number the lower the priority. ?0? stands for the highest priority while ?4294967295? is the lowest.

Default: the lowest priority

To make the configuration simple and reduce the amount of configuration options the ?files? provider has some special properties:

? if maprule is not set the RULE_NAME name is assumed to be the name of the matching user

? if a maprule is used both a single user name or a template like ?{subject_rfc822_name.short_name}? must be in braces like e.g. ?(username)? or ?({subject_rfc822_name.short_name})?

? the ?domains? option is ignored

PROMPTING CONFIGURATION SECTION

If a special file (/var/lib/sss/pubconf/pam_preauth_available) exists SSSD's PAM module pam_sss will ask SSSD to figure out which authentication methods are available for the user trying to log in. Based on the results pam_sss will prompt the user for appropriate credentials.

With the growing number of authentication methods and the possibility that there are multiple ones for a single user the heuristic used by pam_sss to select the prompting

might not be suitable for all use cases. The following options should provide a better flexibility here.

Each supported authentication method has its own configuration subsection under ?[prompting/...]?. Currently there are:

[prompting/password]

to configure password prompting, allowed options are:

password_prompt

to change the string of the password prompt

[prompting/2fa]

to configure two-factor authentication prompting, allowed options are:

first_prompt

to change the string of the prompt for the first factor

second_prompt

to change the string of the prompt for the second factor

single_prompt

boolean value, if True there will be only a single prompt using the value of first_prompt where it is expected that both factors are entered as a single string. Please note that both factors have to be entered here, even if the second factor is optional.

If the second factor is optional and it should be possible to log in either only with the password or with both factors two-step prompting has to be used.

It is possible to add a subsection for specific PAM services, e.g. ?[prompting/password/sshd]? to individual change the prompting for this service.

EXAMPLES

1. 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
```

2. The following example shows configuration of IPA AD trust where the AD forest consists of two domains in a parent-child structure. Suppose IPA domain (ipa.com) has trust with AD domain(ad.com). ad.com has child domain (child.ad.com). To enable shortnames in the child domain the following configuration should be used.

```
[domain/ipa.com/child.ad.com]
use_fully_qualified_names = false
```

3. The following example shows the configuration for two certificate mapping rules. The first is valid for the configured domain ?my.domain? and additionally for the subdomains ?your.domain? and uses the full certificate in the search filter. The second example is valid for the domain ?files? where it is assumed the files provider is used for this domain and contains a matching rule for the local user ?myname?.

```
[certmap/my.domain/rule_name]
matchrule = <ISSUER>^CN=My-CA,DC=MY,DC=DOMAIN$
maprule = (userCertificate;binary={cert!bin})
domains = my.domain, your.domain
priority = 10
[certmap/files/myname]
matchrule = <ISSUER>^CN=My-CA,DC=MY,DC=DOMAIN$<SUBJECT>^CN=User.Name,DC=MY,DC=DOMAIN$
```

SEE ALSO

sssd(8), sssd.conf(5), sssd-ldap(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 10/04/2022 SSSD.CONF(5)