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

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

$ man sssd-ad.5

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

NAME

sssd-ad - SSSD Active Directory provider

DESCRIPTION

This manual page describes the configuration of the AD provider for sssd(8). For a detailed syntax reference, refer to the ?FILE FORMAT? section of the sssd.conf(5) manual page.

The AD provider is a back end used to connect to an Active Directory server. This provider requires that the machine be joined to the AD domain and a keytab is available. Back end communication occurs over a GSSAPI-encrypted channel, SSL/TLS options should not be used with the AD provider and will be superseded by Kerberos usage.

The AD provider supports connecting to Active Directory 2008 R2 or later. Earlier versions may work, but are unsupported.

The AD provider can be used to get user information and authenticate users from trusted domains. Currently only trusted domains in the same forest are recognized. In addition servers from trusted domains are always auto-discovered.

The AD provider enables SSSD to use the sssd-ldap(5) identity provider and the sssd-krb5(5) authentication provider with optimizations for Active Directory environments. The AD provider accepts the same options used by the sssd-ldap and sssd-krb5 providers with some exceptions. However, it is neither necessary nor recommended to set these options.

The AD provider primarily copies the traditional ldap and krb5 provider default options with some exceptions, the differences are listed in the ?MODIFIED DEFAULT OPTIONS? section.

The AD provider can also be used as an access, chpass, sudo and autofs provider. No configuration of the access provider is required on the client side.

If ?auth_provider=ad? or ?access_provider=ad? is configured in sssd.conf then the id_provider must also be set to ?ad?.

By default, the AD provider will map UID and GID values from the objectSID parameter in Active Directory. For details on this, see the ?ID MAPPING? section below. If you want to disable ID mapping and instead rely on POSIX attributes defined in Active Directory, you should set

    ldap_id_mapping = False

If POSIX attributes should be used, it is recommended for performance reasons that the attributes are also replicated to the Global Catalog.

If POSIX attributes are replicated, SSSD will attempt to locate the domain of a requested numerical ID with the help of the Global Catalog and only search that domain. In contrast, if POSIX attributes are not replicated to the Global Catalog, SSSD must search all the domains in the forest sequentially. Please note that the ?cache_first? option might be also helpful in speeding up domainless searches. Note that if only a subset of POSIX attributes is present in the Global Catalog, the non-replicated attributes are currently not read from the LDAP port.

Users, groups and other entities served by SSSD are always treated as case-insensitive in the AD provider for compatibility with Active Directory's LDAP implementation.

SSSD only resolves Active Directory Security Groups. For more information about AD group types see: Active Directory security groups[1]

SSSD filters out Domain Local groups from remote domains in the AD forest. By default they are filtered out e.g. when following a nested group hierarchy in remote domains because they are not valid in the

local domain. This is done to be in agreement with Active Directory's group-membership assignment which can be seen in the PAC of the Kerberos ticket of a user issued by Active Directory.

CONFIGURATION OPTIONS

Refer to the section ?DOMAIN SECTIONS? of the sssd.conf(5) manual page for details on the configuration of an SSSD domain.

ad_domain (string)

Specifies the name of the Active Directory domain. This is optional. If not provided, the configuration domain name is used.

For proper operation, this option should be specified as the lower-case version of the long version of the Active Directory domain.

The short domain name (also known as the NetBIOS or the flat name) is autodetected by the SSSD.

ad_enabled_domains (string)

A comma-separated list of enabled Active Directory domains. If provided, SSSD will ignore any domains not listed in this option.

If left unset, all domains from the AD forest will be available.

For proper operation, this option must be specified in all lower-case and as the fully qualified domain name of the Active Directory domain. For example:

ad_enabled_domains = sales.example.com, eng.example.com

The short domain name (also known as the NetBIOS or the flat name) will be autodetected by SSSD.

Default: Not set

ad_server, ad_backup_server (string)

The comma-separated list of hostnames of the AD servers to which SSSD should connect in order of preference. For more information on failover and server redundancy, see the ?FAILOVER? section.

This is optional if autodiscovery is enabled. For more information on service discovery, refer to the ?SERVICE DISCOVERY? section.

Note: Trusted domains will always auto-discover servers even if the primary server is explicitly defined in the ad_server option.

ad_hostname (string)

Optional. On machines where the hostname(5) does not reflect the fully qualified name, sssd will try to expand the short name. If it is not possible or the short name should be really used instead, set this parameter explicitly.

This field is used to determine the host principal in use in the keytab and to perform dynamic DNS updates. It must match the hostname for which the keytab was issued.

ad_enable_dns_sites (boolean)

Enables DNS sites - location based service discovery.

If true and service discovery (see Service Discovery paragraph at the bottom of the man page) is enabled, the SSSD will first attempt to discover the Active Directory server to connect to using the Active Directory Site Discovery and fall back to the DNS SRV records if no AD site is found. The DNS SRV configuration, including the discovery domain, is used during site discovery as well.

Default: true

ad_access_filter (string)

This option specifies LDAP access control filter that the user must match in order to be allowed access. Please note that the ?access_provider? option must be explicitly set to ?ad? in order for this option to have an effect.

The option also supports specifying different filters per domain or forest. This extended filter would consist of: ?KEYWORD:NAME:FILTER?. The keyword can be either ?DOM?, ?FOREST? or missing.

If the keyword equals to ?DOM? or is missing, then ?NAME? specifies the domain or subdomain the filter applies to. If the keyword equals to ?FOREST?, then the filter equals to all domains from the forest specified by ?NAME?.

Multiple filters can be separated with the ??? character, similarly to how search bases work.

Nested group membership must be searched for using a special OID ?:1.2.840.113556.1.4.1941:? in addition to the full DOM:domain.example.org: syntax to ensure the parser does not attempt to interpret the colon characters associated with the OID. If you do not use this OID then nested group membership will not be resolved. See usage example below and refer here for further information about the OID: [MS-ADTS] section LDAP extensions[2] The most specific match is always used. For example, if the option specified filter for a domain the user is a member of and a global filter, the per-domain filter would be applied. If there are more matches with the same specification, the first one is used.

Examples:

apply filter on domain called dom1 only:

dom1:(memberOf=cn=admins,ou=groups,dc=dom1,dc=com)

apply filter on domain called dom2 only:

DOM:dom2:(memberOf=cn=admins,ou=groups,dc=dom2,dc=com)

apply filter on forest called EXAMPLE.COM only:

FOREST:EXAMPLE.COM:(memberOf=cn=admins,ou=groups,dc=example,dc=com)

apply filter for a member of a nested group in dom1:

DOM:dom1:(memberOf:1.2.840.113556.1.4.1941:=cn=nestedgroup,ou=groups,dc=example,dc=com)

Default: Not set

ad_site (string)

Specify AD site to which client should try to connect. If this option is not provided, the AD site will be auto-discovered.

Default: Not set

ad_enable_gc (boolean)

By default, the SSSD connects to the Global Catalog first to retrieve users from trusted domains and uses the LDAP port to retrieve group memberships or as a fallback. Disabling this option makes the SSSD only connect to the LDAP port of the current AD server.

Please note that disabling Global Catalog support does not disable retrieving users from trusted domains. The SSSD would connect to

the LDAP port of trusted domains instead. However, Global Catalog must be used in order to resolve cross-domain group memberships.

Default: true

ad_gpo_access_control (string)

This option specifies the operation mode for GPO-based access control functionality: whether it operates in disabled mode, enforcing mode, or permissive mode. Please note that the ?access_provider? option must be explicitly set to ?ad? in order for this option to have an effect.

GPO-based access control functionality uses GPO policy settings to determine whether or not a particular user is allowed to logon to the host. For more information on the supported policy settings please refer to the ?ad_gpo_map? options.

Please note that current version of SSSD does not support Active Directory's built-in groups. Built-in groups (such as Administrators with SID S-1-5-32-544) in GPO access control rules will be ignored by SSSD. See upstream issue tracker https://github.com/SSSD/sssd/issues/5063 .

Before performing access control SSSD applies group policy security filtering on the GPOs. For every single user login, the applicability of the GPOs that are linked to the host is checked.

In order for a GPO to apply to a user, the user or at least one of the groups to which it belongs must have following permissions on the GPO:

? Read: The user or one of its groups must have read access to the properties of the GPO (RIGHT_DS_READ_PROPERTY)

? Apply Group Policy: The user or at least one of its groups must be allowed to apply the GPO (RIGHT_DS_CONTROL_ACCESS).

By default, the Authenticated Users group is present on a GPO and this group has both Read and Apply Group Policy access rights.

Since authentication of a user must have been completed successfully before GPO security filtering and access control are started, the Authenticated Users group permissions on the GPO

always apply also to the user.

NOTE: If the operation mode is set to enforcing, it is possible that users that were previously allowed logon access will now be denied logon access (as dictated by the GPO policy settings). In order to facilitate a smooth transition for administrators, a permissive mode is available that will not enforce the access control rules, but will evaluate them and will output a syslog message if access would have been denied. By examining the logs, administrators can then make the necessary changes before setting the mode to enforcing. For logging GPO-based access control debug level 'trace functions' is required (see sssctl(8) manual page).

There are three supported values for this option:

? disabled: GPO-based access control rules are neither evaluated nor enforced.

? enforcing: GPO-based access control rules are evaluated and enforced.

? permissive: GPO-based access control rules are evaluated, but not enforced. Instead, a syslog message will be emitted indicating that the user would have been denied access if this option's value were set to enforcing.

Default: enforcing

ad_gpo_implicit_deny (boolean)

Normally when no applicable GPOs are found the users are allowed access. When this option is set to True users will be allowed access only when explicitly allowed by a GPO rule. Otherwise users will be denied access. This can be used to harden security but be careful when using this option because it can deny access even to users in the built-in Administrators group if no GPO rules apply to them.

Default: False

The following 2 tables should illustrate when a user is allowed or rejected based on the allow and deny login rights defined on the server-side and the setting of ad_gpo_implicit_deny.

??

?ad_gpo_implicit_deny = False (default) ?

??

?allow-rules ? deny-rules ? results ?

??

? missing ? missing ? all users are ?

? ? ? allowed ?

??

? missing ? present ? only users not in ?

? ? ? deny-rules are ?

? ? ? allowed ?

??

? present ? missing ? only users in ?

? ? ? allow-rules are ?

? ? ? allowed ?

??

? present ? present ? only users in ?

? ? ? allow-rules and not ?

? ? ? in deny-rules are ?

? ? ? allowed ?

??

??

?ad_gpo_implicit_deny = True ?

??

?allow-rules ? deny-rules ? results ?

??

? missing ? missing ? no users are ?

? ? ? allowed ?

??

? missing ? present ? no users are ?

? ? ? allowed ?

??

? present ? missing ? only users in ?

? ? ? allow-rules are ?

? ? ? allowed ?

??

? present ? present ? only users in ?

? ? ? allow-rules and not ?

? ? ? in deny-rules are ?

? ? ? allowed ?

??

ad_gpo_ignore_unreadable (boolean)

Normally when some group policy containers (AD object) of applicable group policy objects are not readable by SSSD then users are denied access. This option allows to ignore group policy containers and with them associated policies if their attributes in group policy containers are not readable for SSSD.

Default: False

ad_gpo_cache_timeout (integer)

The amount of time between lookups of GPO policy files against the AD server. This will reduce the latency and load on the AD server if there are many access-control requests made in a short period.

Default: 5 (seconds)

ad_gpo_map_interactive (string)

A comma-separated list of PAM service names for which GPO-based access control is evaluated based on the InteractiveLogonRight and DenyInteractiveLogonRight policy settings. Only those GPOs are evaluated for which the user has Read and Apply Group Policy permission (see option ?ad_gpo_access_control?). If an evaluated GPO contains the deny interactive logon setting for the user or one of its groups, the user is denied local access. If none of the evaluated GPOs has an interactive logon right defined, the user is granted local access. If at least one evaluated GPO contains interactive logon right settings, the user is granted local access only, if it or at least one of its groups is part of the policy settings.

Note: Using the Group Policy Management Editor this value is called "Allow log on locally" and "Deny log on locally".

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 this logon right (e.g. ?login?) with a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_interactive = +my_pam_service, -login

Default: the default set of PAM service names includes:

- login
- su
- su-l
- gdm-fingerprint
- gdm-password
- gdm-smartcard
- kdm
- lightdm
- lxdm
- sddm
- unity
- xdm

ad_gpo_map_remote_interactive (string)

A comma-separated list of PAM service names for which GPO-based access control is evaluated based on the RemoteInteractiveLogonRight and DenyRemoteInteractiveLogonRight policy settings. Only those GPOs are evaluated for which the user has Read and Apply Group Policy permission (see option ?ad_gpo_access_control?). If an evaluated GPO contains the deny remote logon setting for the user or one of its groups, the user is denied remote interactive access. If none of the evaluated GPOs has a remote interactive logon right defined, the user is granted remote access. If at least one evaluated GPO contains remote

interactive logon right settings, the user is granted remote access only, if it or at least one of its groups is part of the policy settings.

Note: Using the Group Policy Management Editor this value is called "Allow log on through Remote Desktop Services" and "Deny log on through Remote Desktop Services".

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 this logon right (e.g. ?sshd?) with a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_remote_interactive = +my_pam_service, -sshd

Default: the default set of PAM service names includes:

? sshd

? cockpit

ad_gpo_map_network (string)

A comma-separated list of PAM service names for which GPO-based access control is evaluated based on the NetworkLogonRight and DenyNetworkLogonRight policy settings. Only those GPOs are evaluated for which the user has Read and Apply Group Policy permission (see option ?ad_gpo_access_control?). If an evaluated GPO contains the deny network logon setting for the user or one of its groups, the user is denied network logon access. If none of the evaluated GPOs has a network logon right defined, the user is granted logon access. If at least one evaluated GPO contains network logon right settings, the user is granted logon access only, if it or at least one of its groups is part of the policy settings.

Note: Using the Group Policy Management Editor this value is called "Access this computer from the network" and "Deny access to this computer from the network".

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 this logon right (e.g. ?ftp?) with a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_network = +my_pam_service, -ftp

Default: the default set of PAM service names includes:

? ftp

? samba

ad_gpo_map_batch (string)

A comma-separated list of PAM service names for which GPO-based access control is evaluated based on the BatchLogonRight and DenyBatchLogonRight policy settings. Only those GPOs are evaluated for which the user has Read and Apply Group Policy permission (see option ?ad_gpo_access_control?). If an evaluated GPO contains the deny batch logon setting for the user or one of its groups, the user is denied batch logon access. If none of the evaluated GPOs has a batch logon right defined, the user is granted logon access. If at least one evaluated GPO contains batch logon right settings, the user is granted logon access only, if it or at least one of its groups is part of the policy settings.

Note: Using the Group Policy Management Editor this value is called "Allow log on as a batch job" and "Deny log on as a batch job".

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 this logon right (e.g. ?crond?) with a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_batch = +my_pam_service, -crond

Note: Cron service name may differ depending on Linux distribution used.

Default: the default set of PAM service names includes:

? crond

ad_gpo_map_service (string)

A comma-separated list of PAM service names for which GPO-based access control is evaluated based on the ServiceLogonRight and DenyServiceLogonRight policy settings. Only those GPOs are evaluated for which the user has Read and Apply Group Policy permission (see option ?ad_gpo_access_control?). If an evaluated GPO contains the deny service logon setting for the user or one of its groups, the user is denied service logon access. If none of the evaluated GPOs has a service logon right defined, the user is granted logon access. If at least one evaluated GPO contains service logon right settings, the user is granted logon access only, if it or at least one of its groups is part of the policy settings.

Note: Using the Group Policy Management Editor this value is called "Allow log on as a service" and "Deny log on as a service".

It is possible to add a PAM service name to the default set by using ?+service_name?. Since the default set is empty, it is not possible to remove a PAM service name from the default set. For example, in order to add a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_service = +my_pam_service

Default: not set

ad_gpo_map_permit (string)

A comma-separated list of PAM service names for which GPO-based access is always granted, regardless of any GPO Logon Rights.

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 unconditionally permitted access (e.g. ?sudo?) with a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_permit = +my_pam_service, -sudo

Default: the default set of PAM service names includes:

? polkit-1

? sudo

? sudo-i

? systemd-user

ad_gpo_map_deny (string)

A comma-separated list of PAM service names for which GPO-based access is always denied, regardless of any GPO Logon Rights.

It is possible to add a PAM service name to the default set by using ?+service_name?. Since the default set is empty, it is not possible to remove a PAM service name from the default set. For example, in order to add a custom pam service name (e.g. ?my_pam_service?), you would use the following configuration:

ad_gpo_map_deny = +my_pam_service

Default: not set

ad_gpo_default_right (string)

This option defines how access control is evaluated for PAM service names that are not explicitly listed in one of the ad_gpo_map_* options. This option can be set in two different manners. First, this option can be set to use a default logon right. For example, if this option is set to 'interactive', it means that unmapped PAM service names will be processed based on the InteractiveLogonRight and DenyInteractiveLogonRight policy settings. Alternatively, this option can be set to either always permit or always deny access for unmapped PAM service names.

Supported values for this option include:

? interactive

? remote_interactive

? network

? batch

? service

? permit

? deny

Default: deny

ad_maximum_machine_account_password_age (integer)

SSSD will check once a day if the machine account password is older than the given age in days and try to renew it. A value of 0 will disable the renewal attempt.

Default: 30 days

ad_machine_account_password_renewal_opts (string)

This option should only be used to test the machine account renewal task. The option expects 2 integers separated by a colon (':'). The first integer defines the interval in seconds how often the task is run. The second specifies the initial timeout in seconds before the task is run for the first time after startup.

Default: 86400:750 (24h and 15m)

ad_update_samba_machine_account_password (boolean)

If enabled, when SSSD renews the machine account password, it will also be updated in Samba's database. This prevents Samba's copy of the machine account password from getting out of date when it is set up to use AD for authentication.

Default: false

ad_use_ldaps (bool)

By default SSSD uses the plain LDAP port 389 and the Global Catalog port 3628. If this option is set to True SSSD will use the LDAPS port 636 and Global Catalog port 3629 with LDAPS protection. Since AD does not allow to have multiple encryption layers on a single connection and we still want to use SASL/GSSAPI or SASL/GSS-SPNEGO for authentication the SASL security property maxssf is set to 0 (zero) for those connections.

Default: False

ad_allow_remote_domain_local_groups (boolean)

If this option is set to ?true? SSSD will not filter out Domain Local groups from remote domains in the AD forest. By default they are filtered out e.g. when following a nested group hierarchy in

remote domains because they are not valid in the local domain. To be compatible with other solutions which make AD users and groups available on Linux client this option was added.

Please note that setting this option to ?true? will be against the intention of Domain Local group in Active Directory and SHOULD ONLY BE USED TO FACILITATE MIGRATION FROM OTHER SOLUTIONS. Although the group exists and user can be member of the group the intention is that the group should be only used in the domain it is defined and in no others. Since there is only one type of POSIX groups the only way to achieve this on the Linux side is to ignore those groups. This is also done by Active Directory as can be seen in the PAC of the Kerberos ticket for a local service or in tokenGroups requests where remote Domain Local groups are missing as well.

Given the comments above, if this option is set to ?true? the tokenGroups request must be disabled by setting ?ldap_use_tokengroups? to ?false? to get consistent group-memberships of a users. Additionally the Global Catalog lookup should be skipped as well by setting ?ad_enable_gc? to ?false?. Finally it might be necessary to modify ?ldap_group_nesting_level? if the remote Domain Local groups can only be found with a deeper nesting level.

Default: False

dyndns_update (boolean)

Optional. This option tells SSSD to automatically update the Active Directory DNS server with the IP address of this client. The update is secured using GSS-TSIG. As a consequence, the Active Directory administrator only needs to allow secure updates for the DNS zone. The IP address of the AD LDAP connection is used for the updates, if it is not otherwise specified by using the ?dyndns_iface? option.

NOTE: On older systems (such as RHEL 5), for this behavior to work reliably, the default Kerberos realm must be set properly in /etc/krb5.conf

Default: true

dyndns_ttl (integer)

The TTL to apply to the client DNS record when updating it. If dyndns_update is false this has no effect. This will override the TTL serverside if set by an administrator.

Default: 3600 (seconds)

dyndns_iface (string)

Optional. Applicable only when dyndns_update is true. Choose the interface or a list of interfaces whose IP addresses should be used for dynamic DNS updates. Special value ?*? implies that IPs from all interfaces should be used.

Default: Use the IP addresses of the interface which is used for AD LDAP connection

Example: dyndns_iface = em1, vnet1, vnet2

dyndns_refresh_interval (integer)

How often should the back end perform periodic DNS update in addition to the automatic update performed when the back end goes online. This option is optional and applicable only when dyndns_update is true. Note that the lowest possible value is 60 seconds in-case if value is provided less than 60, parameter will assume lowest value only.

Default: 86400 (24 hours)

dyndns_update_ptr (bool)

Whether the PTR record should also be explicitly updated when updating the client's DNS records. Applicable only when dyndns_update is true.

Default: True

dyndns_force_tcp (bool)

Whether the nsupdate utility should default to using TCP for communicating with the DNS server.

Default: False (let nsupdate choose the protocol)

dyndns_auth (string)

Whether the nsupdate utility should use GSS-TSIG authentication for

secure updates with the DNS server, insecure updates can be sent by setting this option to 'none'.

Default: GSS-TSIG

dyndns_auth_ptr (string)

Whether the nsupdate utility should use GSS-TSIG authentication for secure PTR updates with the DNS server, insecure updates can be sent by setting this option to 'none'.

Default: Same as dyndns_auth

dyndns_server (string)

The DNS server to use when performing a DNS update. In most setups, it's recommended to leave this option unset.

Setting this option makes sense for environments where the DNS server is different from the identity server.

Please note that this option will be only used in fallback attempt when previous attempt using autodetected settings failed.

Default: None (let nsupdate choose the server)

dyndns_update_per_family (boolean)

DNS update is by default performed in two steps - IPv4 update and then IPv6 update. In some cases it might be desirable to perform IPv4 and IPv6 update in single step.

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 original home directory retrieved from the identity provider, but in lower case.

%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

krb5_confd_path (string)

Absolute path of a directory where SSSD should place Kerberos

configuration snippets.

To disable the creation of the configuration snippets set the parameter to 'none'.

Default: not set (krb5.include.d subdirectory of SSSD's pubconf directory)

MODIFIED DEFAULT OPTIONS

Certain option defaults do not match their respective backend provider defaults, these option names and AD provider-specific defaults are listed below:

KRB5 Provider

- krb5_validate = true
- krb5_use_enterprise_principal = true

LDAP Provider

- ldap_schema = ad
- ldap_force_upper_case_realm = true
- ldap_id_mapping = true
- ldap_sasl_mech = GSS-SPNEGO
- ldap_referrals = false
- ldap_account_expire_policy = ad
- ldap_use_tokengroups = true
- ldap_sasl_authid = sAMAccountName@REALM (typically SHORTNAME$@REALM)

  The AD provider looks for a different principal than the LDAP provider by default, because in an Active Directory environment the principals are divided into two groups - User Principals and Service Principals. Only User Principal can be used to obtain a TGT and by default, computer object's principal is constructed from its sAMAccountName and the AD realm. The well-known host/hostname@REALM principal is a Service Principal and thus cannot be used to get a TGT with.

NSS configuration

- fallback_homedir = /home/%d/%u

  The AD provider automatically sets "fallback_homedir = /home/%d/%u"

to provide personal home directories for users without the homeDirectory attribute. If your AD Domain is properly populated with Posix attributes, and you want to avoid this fallback behavior, you can explicitly set "fallback_homedir = %o".

Note that the system typically expects a home directory in /home/%u folder. If you decide to use a different directory structure, some other parts of your system may need adjustments.

For example automated creation of home directories in combination with selinux requires selinux adjustment, otherwise the home directory will be created with wrong selinux context.

FAILOVER

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

Failover Syntax

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

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

The Failover Mechanism

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

is still considered online and might still be tried for another service.

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

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

Failover time outs and tuning

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

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

dns_resolver_server_timeout

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

Default: 1000

dns_resolver_op_timeout

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

Default: 3

dns_resolver_timeout

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

Default: 6

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

SERVICE DISCOVERY

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

Configuration

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

The domain name

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

The protocol

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

See Also

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

ID MAPPING

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

NOTE: When ID-mapping is enabled, the uidNumber and gidNumber attributes are ignored. This is to avoid the possibility of conflicts

between automatically-assigned and manually-assigned values. If you need to use manually-assigned values, ALL values must be manually-assigned.

Please note that changing the ID mapping related configuration options will cause user and group IDs to change. At the moment, SSSD does not support changing IDs, so the SSSD database must be removed. Because cached passwords are also stored in the database, removing the database should only be performed while the authentication servers are reachable, otherwise users might get locked out. In order to cache the password, an authentication must be performed. It is not sufficient to use sss_cache(8) to remove the database, rather the process consists of:

? Making sure the remote servers are reachable

? Stopping the SSSD service

? Removing the database

? Starting the SSSD service

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

Mapping Algorithm

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

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

When a user or group entry for a particular domain is encountered for the first time, the SSSD allocates one of the available slices for that domain. In order to make this slice-assignment repeatable on different client machines, we select the slice based on the following algorithm:

The SID string is passed through the murmurhash3 algorithm to convert it to a 32-bit hashed value. We then take the modulus of this value with the total number of available slices to pick the slice.

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

Configuration

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

ldap_id_mapping = True

ldap_schema = ad

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

Advanced Configuration

ldap_idmap_range_min (integer)

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

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

Default: 200000

ldap_idmap_range_max (integer)

Specifies the upper (exclusive) bound of the range of POSIX IDs to use for mapping Active Directory user and group SIDs. It is the first POSIX ID which cannot be used for the mapping

anymore, i.e. one larger than the last one which can be used for the mapping.

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

Default: 2000200000

ldap_idmap_range_size (integer)

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

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

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

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

Default: 200000

ldap_idmap_default_domain_sid (string)

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

Default: not set

ldap_idmap_default_domain (string)

Specify the name of the default domain.

Default: not set

ldap_idmap_autorid_compat (boolean)

Changes the behavior of the ID-mapping algorithm to behave more similarly to winbind's ?idmap_autorid? algorithm.

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

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

Default: False

ldap_idmap_helper_table_size (integer)

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

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

Default: 10

Well-Known SIDs

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

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

? Null Authority

? World Authority

? Local Authority

? Creator Authority

? Mandatory Label Authority

? Authentication Authority

? NT Authority

? Built-in

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

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

EXAMPLE

The following example assumes that SSSD is correctly configured and example.com is one of the domains in the [sssd] section. This example shows only the AD provider-specific options.

```
[domain/EXAMPLE]
id_provider = ad
auth_provider = ad
access_provider = ad
chpass_provider = ad
ad_server = dc1.example.com
ad_hostname = client.example.com
ad_domain = example.com
```

NOTES

The AD access control provider checks if the account is expired. It has the same effect as the following configuration of the LDAP provider:

```
access_provider = ldap
```

ldap_access_order = expire

ldap_account_expire_policy = ad

However, unless the ?ad? access control provider is explicitly configured, the default access provider is ?permit?. Please note that if you configure an access provider other than ?ad?, you need to set all the connection parameters (such as LDAP URIs and encryption details) manually.

When the autofs provider is set to ?ad?, the RFC2307 schema attribute mapping (nisMap, nisObject, ...) is used, because these attributes are included in the default Active Directory schema.

SEE ALSO

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

AUTHORS

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

NOTES

1. Active Directory security groups

   https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-groups

2. [MS-ADTS] section LDAP extensions

   https://msdn.microsoft.com/en-us/library/cc223367.aspx

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