Explain - System Security Services Daemon (SSSD) Config.

The System Security Services Daemon (SSSD) provides a set of daemons to manage access to remote directories and authentication mechanisms. It provides Name Service Switch (NSS) and Pluggable Authentication Modules(PAM) interfaces toward the system and a pluggable back end system to connect to multiple different account sources.

Table of Contents

Here is our configuration.

```
#################################################
# SSSD Configuration
#################################################

[sssd]
config_file_version = 2
debug_level = 0
domains = user-account.domain.com
services = nss, pam

[nss]
filter_groups = root
filter_users = root
reconnection_retries = 3
entry_cache_timeout = 3
entry_cache_nowait_percentage = 75
debug_level = 8
account_cache_expiration = 1

[pam]
reconnection_retries = 3

#################################################
# `user-account.domain.com` Configuration
#################################################

[domain/user-account.domain.com]
# Debugging
debug_level = 1

# id and auth provider
id_provider = ldap
auth_provider = ldap



# Access filter
access_provider = ldap
ldap_access_filter = (&(objectClass=user) \
                        (memberof:1.2.840.113556.1.4.1941:=cn=some_user_group_in_as_genesis,\
                             ou=servergroups,ou=accessmgmnt,dc=user-account,dc=domain,dc=com))

# Caching. Enabling Offline Authentication.
cache_credentials = true
entry_cache_timeout = 3

# Min ID and AD information.
min_id = 1000
ad_server = user-auth-srv-pri.user-account.domain.com,user-auth-srv-sec.user-account.domain.com

# Ldap Details.
ldap_uri = ldaps://ldap.user-account.domain.com
ldap_schema = ad
ldap_id_mapping = true
ldap_referrals = false
```

```
ldap_default_bind_dn = cn=svc-genesis-ldapbind,ou=serviceaccounts,
                                ou=accounts,ou=accessmgmnt,dc=user-account,dc=domain,dc=com
ldap_default_authtok_type = password
ldap_default_authtok = some_random_password_as_in_ad
fallback_homedir = /home/%u
ldap_user_home_directory = unixHomeDirectory
ldap_tls_cacert = /etc/openldap/cacerts/ssl-cacerts.cer
ignore_group_members = true

# krb5 details.
krb5_realm = USER-ACCOUNT.DOMAIN.COM
krb5_server =  user-auth-srv-pri.user-account.domain.com,user-auth-srv-sec.user-account.domain.com
chpass_provider = krb5
```

Configuring Services: PAM

Configuring Services: PAM

SSSD provides a PAM module, sssd_pam, which instructs the system to use SSSD to retrieve user information. The PAM configuration must include a reference to the SSSD module, and then the SSSD configuration sets how SSSD interacts with PAM.

```
# authconfig --update --enablesssd --enablesssdauth
```

This automatically updates the PAM configuration to reference all of the SSSD modules:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_sss.so use_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password sufficient pam_sss.so use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session sufficient pam_sss.so
session     required      pam_unix.so
```

SSSD and Identity Providers (Domains)

SSSD recognizes domains, which are entries within the SSSD configuration file associated with different, external data sources. Domains are a combination of an identity provider (for user information) and, optionally, other providers such as authentication (for authentication requests) and for other operations, such as password changes. (The identity provider can also be used for all operations, if all operations are performed within a single domain or server.)

SSSD and Identity Providers

```
# id and auth provider
id_provider = ldap
auth_provider = ldap
```

`id_provider` (string)

Specifies the data back end to use for this domain. The supported identity back ends are:

- `ldap`
- `ipa` (Identity Management in Red Hat Enterprise Linux)
- `ad` (Microsoft Active Directory)
- `proxy`, for a legacy NSS provider, such as `nss_nis`. Using a proxy ID provider also requires specifying the legacy NSS library to load to start successfully, set in the `proxy_lib_name` option.
- `local`, the SSSD internal local provider

`auth_provider` (string)

Sets the authentication provider used for the domain. The default value for this option is the value of id_provider. The supported authentication providers are `ldap`, `ipa`, `ad`, `krb5` (Kerberos), `proxy`, and `none`.

`cache_credentials` (Boolean Optional).

Specifies whether to store user credentials in the local SSSD domain database cache. The default value for this parameter is false. Set this value to true for domains other than the LOCAL domain to enable offline authentication.

Using the Access Filters

An LDAP, Active Directory, or Identity Management server can provide access control rules for adomain. The associated options (`ldap_access_filter` for `LDAP` and `IdM` and `ad_access_filter` for AD) specify which users are granted access to the specified host. The user filter must be used or all users are denied access. See the examples below:

```
# Access filter
access_provider = ldap
ldap_access_filter = (&(objectClass=user)
                        (memberof:1.2.840.113556.1.4.1941:=cn=some_user_group_in_as_genesis,
                                ou=servergroups,ou=accessmgmnt,dc=user-account,dc=domain,dc=com))
```

`access_provider`, `ldap_access_filter`

If using `access_provider` = `ldap`, this option is mandatory. It specifies an LDAP search filter criteria that must be met for the user to be granted access on this host. If `access_provider` = `ldap` and this option is not set, it will result in all users being denied access. Use `access_provider` = allow to change this default behaviour. See more

Cache configuration from `sssd.conf`.

User identities are always cached, as well as information about the domain services. However, user credentials are not cached by default. This means that SSSD always checks with the back end identity provider for authentication requests. If the identity provider is offline or unavailable, there is no way to process those authentication requests, so user authentication could fail.

It is possible to enable offline credentials caching, which stores credentials (after successful login) as part of the user account in the SSSD cache. Therefore, even if an identity provider is unavailable, users can still authenticate, using their stored credentials. Offline credentials caching is primarily configured in each individual domain entry, but there are some optional settings that can be set in the PAM service section, because credentials caching interacts with the local PAM service as well as the remote domain.

Cache configuration

```
# Caching. Enabling Offline Authentication.
cache_credentials = true
entry_cache_timeout = 3
```

`offline_credentials_expiration` sets the number of days after a successful login that a single credentials entry for a user is preserved in cache. Setting this to zero (0) means that entries are kept forever.

While not related to the credentials cache specifically, each domain has configuration options on when individual user and service caches expire:

> `account_cache_expiration` sets the number of days after a successful login that the entire user account entry is removed from the SSSD cache. This must be equal to or longer than the individual offline credentials cache expiration period.
>
> `entry_cache_timeout` sets a validity period, in seconds, for all entries stored in the cache before SSSD requests updated information from the identity provider. There are also individual cache timeout parameters for group, service, netgroup, sudo, and autofs entries; these are listed in the sssd.conf man page. The default time is 5400 seconds (90 minutes).

Here is an Example:

NOTE : Make sure we have the `offline_credentials_expiration` is set for the configuration to work.

```
[sssd]
services = nss,pam
...

[pam]
offline_credentials_expiration = 3
...

[domain/EXAMPLE]
cache_credentials = true
account_cache_expiration = 7
entry_cache_timeout = 14400
...
```

Debugging

Each domain sets its own debug log level. Increasing the log level can provide more information about problems with SSSD or with the domain configuration.

Debugging

Example:

```
[domain/LDAP]
cache_credentials = true
debug_level = 9
```

Here is some more details.

```
0   Fatal failures. Anything that would prevent SSSD from starting up or causes
        it to cease running.
1   Critical failures. An error that doesn't kill the SSSD, but one that indicates that at
        least one major feature is not going to work properly.
2   Serious failures. An error announcing that a particular request or operation has failed.
3   Minor failures. These are the errors that would percolate down to cause the operation
        failure of 2.
4   Configuration settings.
5   Function data.
6   Trace messages for operation functions.
7   Trace messages for internal control functions.
8   Contents of function-internal variables that may be interesting.
9   Extremely low-level tracing information.
```

Active Directory identity provider

This tag, will get identity information from server provider. Currently we are using AD.

When we execute the `id` command on the terminal, `SSSD` service will get the information from the server provided below.

```
# Min ID and AD information.
min_id = 1000
ad_server = user-auth-srv-pri.user-account.domain.com,user-auth-srv-sec.user-account.domain.com
```

`ad_server`

Is the active directory identity provider.

`LDAP` configuration from `sssd.conf`.

Creating Domains: LDAP

```
# Ldap Details.
ldap_uri = ldaps://ldap.user-account.domain.com
ldap_schema = ad
ldap_id_mapping = true
ldap_referrals = false
```

```
ldap_default_bind_dn = cn=svc-genesis-ldapbind,ou=serviceaccounts,\
                           ou=accounts,ou=accessmgmnt,dc=user-account,dc=domain,dc=com
ldap_default_authtok_type = password
ldap_default_authtok = some_random_password_as_in_ad
fallback_homedir = /home/%u
ldap_user_home_directory = unixHomeDirectory
ldap_tls_cacert = /etc/openldap/cacerts/ssl-cacerts.cer
ignore_group_members = true
```

ldap_uri

Gives a comma-separated list of the URIs of the LDAP servers to which SSSD will connect. The list is given in order of preference, so the first server in the list is tried first. Listing additional servers provides failover protection. This can be detected from the DNS SRV records if it is not given.

ldap_schema

Sets what version of schema to use when searching for user entries. This can be rfc2307, rfc2307bis, ad, or ipa.

The default is rfc2307. In RFC 2307, group objects use a multi-valued attribute, memberuid, which lists the names of the users that belong to that group. In RFC 2307bis, group objects use the member attribute, which contains the full distinguished name (DN) of a user or group entry.

RFC 2307bis allows nested groups using the member attribute. Because these different schema use different definitions for group membership, using the wrong LDAP schema with SSSD can affect both viewing and managing network resources, even if the appropriate permissions are in place.

For example, with RFC 2307bis, all groups are returned when using nested groups or primary/secondary groups.

```
$ id
uid=500(myserver) gid=500(myserver) groups=500(myserver),510(myothergroup)
```

If SSSD is using RFC 2307 schema, only the primary group is returned. This setting only affects how SSSD determines the group members. It does not change the actual user data.

ldap_tls_reqcert

Specifies how to check for SSL server certificates in a TLS session. There are four options:

- `never` disables requests for certificates.
- `allow` requests a certificate, but proceeds normally even if no certificate is given or a bad certificate is given.
- `try` requests a certificate and proceeds normally if no certificate is given, If a bad certificate is given, the session terminates.
- `demand` and `hard` are the same option. This requires a valid certificate or the session is terminated.

The default is `hard`.

ldap_tls_cacert

Gives the full path and file name to the file that contains the CA certificates for all of the CAs that SSSD recognizes. SSSD will accept any certificate issued by these CAs. This uses the OpenLDAP system defaults if it is not given explicitly.

ldap_referrals

Sets whether SSSD will use LDAP referrals, meaning forwarding queries from one LDAP database to another. SSSD supports database-level and subtree referrals. For referrals within the same LDAP server, SSSD will adjust the DN of the entry being queried. For referrals that go to different LDAP servers, SSSD does an exact match on the DN. Setting this value to true enables referrals; this is the default. Referrals can negatively impact overall performance because of the time spent attempting to trace referrals. Disabling referral checking can significantly improve performance.

ldap_default_bind_dn, ldap_default_authtok_type, ldap_default_authtok

Here is the snippet of the configuration.

```
ldap_default_bind_dn = cn=svc-genesis-ldapbind,ou=serviceaccounts,\
                           ou=accounts,ou=accessmgmnt,dc=user-account,dc=domain,dc=com
ldap_default_authtok_type = password
ldap_default_authtok = some_random_password_as_in_ad
```

We are using these to authenticate with the domain. We need to create a bind user in the active directory domain, so that we can search for the user in the domain.

SSSD is going to authenticate it self with AD, so that we can search in the domain.

ignore_group_members (bool)

Configuration Man Page

Do not return group members for group lookups.

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

Default: `FALSE`

krb5 configuration from sssd.conf.

Both LDAP and proxy identity providers can use a separate Kerberos domain to supply authentication. Configuring a Kerberos authentication provider requires the `key distribution center` (KDC) and the Kerberos domain. All of the principal names must be available in the specified identity provider; if they are not, SSSD constructs the principals using the format `username@REALM`.

NOTE : Kerberos can only provide authentication; it cannot provide an identity database.

krb5 configuration

```
# krb5 details.
chpass_provider = krb5
krb5_realm = USER-ACCOUNT.DOMAIN.COM
krb5_server =  user-auth-srv-pri.user-account.domain.com,user-auth-srv-sec.user-account.domain.com
```

chpass_provider

Specifies which service to use for password change operations. This is assumed to be the same as the authentication provider. To use Kerberos, set this to krb5.

`krb5_server`

Gives the primary Kerberos server, by IP address or host names, to which SSSD will connect.

`krb5_realm`

Identifies the Kerberos realm served by the KDC.