NAME

sssd-ipa − SSSD IPA provider

DESCRIPTION

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

The IPA provider is a back end used to connect to an IPA server. (Refer to the freeipa.org web site for information about IPA servers.) This provider requires that the machine be joined to the IPA domain; configuration is almost entirely self−discovered and obtained directly from the server.

The IPA provider accepts the same options used by the **sssd-ldap**(5) identity provider and the **sssd-krb5**(5) authentication provider with some exceptions described below.

However, it is neither necessary nor recommended to set these options. IPA provider can also be used as an access and chpass provider. As an access provider it uses HBAC (host−based access control) rules. Please refer to freeipa.org for more information about HBAC. No configuration of access provider is required on the client side.

The IPA provider will use the PAC responder if the Kerberos tickets of users from trusted realms contain a PAC. To make configuration easier the PAC responder is started automatically if the IPA ID provider is configured.

CONFIGURATION OPTIONS

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

ipa_domain (string)
: Specifies the name of the IPA domain. This is optional. If not provided, the configuration domain name is used.

ipa_server, ipa_backup_server (string)
: The comma−separated list of IP addresses or hostnames of the IPA servers to which SSSD should connect in the 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.

ipa_hostname (string)
: Optional. May be set on machines where the hostname(5) does not reflect the fully qualified name used in the IPA domain to identify this host.

dyndns_update (boolean)
: Optional. This option tells SSSD to automatically update the DNS server built into FreeIPA v2 with the IP address of this client. The update is secured using GSS−TSIG. The IP address of the IPA 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

  NOTE: While it is still possible to use the old *ipa_dyndns_update* option, users should migrate to using *dyndns_update* in their config file.

  Default: false

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.

  NOTE: While it is still possible to use the old *ipa_dyndns_ttl* option, users should migrate to using *dyndns_ttl* in their config file.

Default: 1200 (seconds)

dyndns_iface (string)
Optional. Applicable only when dyndns_update is true. Choose the interface whose IP address should be used for dynamic DNS updates.

NOTE: This option currently supports only one interface.

NOTE: While it is still possible to use the old *ipa_dyndns_iface* option, users should migrate to using *dyndns_iface* in their config file.

Default: Use the IP address of the IPA LDAP connection

ipa_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, then the SSSD will first attempt location based discovery using a query that contains "_location.hostname.example.com" and then fall back to traditional SRV discovery. If the location based discovery succeeds, the IPA servers located with the location based discovery are treated as primary servers and the IPA servers located using the traditional SRV discovery are used as back up servers

Default: false

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.

Default: 0 (disabled)

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.

This option should be False in most IPA deployments as the IPA server generates the PTR records automatically when forward records are changed.

Default: False (disabled)

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)

ipa_hbac_search_base (string)
Optional. Use the given string as search base for HBAC related objects.

Default: Use base DN

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

See “ldap_search_base” for information about configuring multiple search bases.

Default: the value of *ldap_search_base*

ipa_selinux_search_base (string)

Optional. Use the given string as search base for SELinux user maps.

See “ldap_search_base” for information about configuring multiple search bases.

Default: the value of *ldap_search_base*

ipa_subdomains_search_base (string)
Optional. Use the given string as search base for trusted domains.

See “ldap_search_base” for information about configuring multiple search bases.

Default: the value of *cn=trusts,%basedn*

ipa_master_domain_search_base (string)
Optional. Use the given string as search base for master domain object.

See “ldap_search_base” for information about configuring multiple search bases.

Default: the value of *cn=ad,cn=etc,%basedn*

ipa_views_search_base (string)
Optional. Use the given string as search base for views containers.

See “ldap_search_base” for information about configuring multiple search bases.

Default: the value of *cn=views,cn=accounts,%basedn*

krb5_validate (boolean)
Verify with the help of krb5_keytab that the TGT obtained has not been spoofed.

Default: true

Note that this default differs from the traditional Kerberos provider back end.

krb5_realm (string)
The name of the Kerberos realm. This is optional and defaults to the value of “ipa_domain”.

The name of the Kerberos realm has a special meaning in IPA − it is converted into the base DN to use for performing LDAP operations.

krb5_canonicalize (boolean)
Specifies if the host and user principal should be canonicalized when connecting to IPA LDAP and also for AS requests. This feature is available with MIT Kerberos >= 1.7

Default: true

krb5_use_fast (string)
Enables flexible authentication secure tunneling (FAST) for Kerberos pre−authentication. The following options are supported:

never use FAST.

try to use FAST. If the server does not support FAST, continue the authentication without it. This is equivalent to not setting this option at all.

demand to use FAST. The authentication fails if the server does not require fast.

Default: try

NOTE: SSSD supports FAST only with MIT Kerberos version 1.8 and later. If SSSD is used with an older version of MIT Kerberos, using this option is a configuration error.

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)

ipa_hbac_refresh (integer)
The amount of time between lookups of the HBAC rules against the IPA server. This will reduce the latency and load on the IPA server if there are many access−control requests made in a short period.

Default: 5 (seconds)

ipa_hbac_selinux (integer)
The amount of time between lookups of the SELinux maps against the IPA server. This will reduce the latency and load on the IPA server if there are many user login requests made in a short period.

Default: 5 (seconds)

ipa_hbac_treat_deny_as (string)
This option specifies how to treat the deprecated DENY−type HBAC rules. As of FreeIPA v2.1, DENY rules are no longer supported on the server. All users of FreeIPA will need to migrate their rules to use only the ALLOW rules. The client will support two modes of operation during this transition period:

DENY_ALL: If any HBAC DENY rules are detected, all users will be denied access.

IGNORE: SSSD will ignore any DENY rules. Be very careful with this option, as it may result in opening unintended access.

Default: DENY_ALL

ipa_server_mode (boolean)
This option should only be set by the IPA installer.

The option denotes that the SSSD is running on IPA server and should perform lookups of users and groups from trusted domains differently.

Default: false

ipa_automount_location (string)
The automounter location this IPA client will be using

Default: The location named "default"

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.

VIEWS AND OVERRIDES

SSSD can handle views and overrides which are offered by FreeIPA 4.1 and later version. Since all paths and objectclasses are fixed on the server side there is basically no need to configure anything. For

completeness the related options are listed here with their default values.

ipa_view_class (string)
Objectclass of the view container.

Default: nsContainer

ipa_view_name (string)
Name of the attribute holding the name of the view.

Default: cn

ipa_overide_object_class (string)
Objectclass of the override objects.

Default: ipaOverrideAnchor

ipa_anchor_uuid (string)
Name of the attribute containing the reference to the original object in a remote domain.

Default: ipaAnchorUUID

ipa_user_override_object_class (string)
Name of the objectclass for user overrides. It is used to determine if the found override object is related to a user or a group.

User overrides can contain attributes given by

- ldap_user_name
- ldap_user_uid_number
- ldap_user_gid_number
- ldap_user_gecos
- ldap_user_home_directory
- ldap_user_shell
- ldap_user_ssh_public_key

Default: ipaUserOverride

ipa_group_override_object_class (string)
Name of the objectclass for group overrides. It is used to determine if the found override object is related to a user or a group.

Group overrides can contain attributes given by

- ldap_group_name
- ldap_group_gid_number

Default: ipaGroupOverride

SUBDOMAINS PROVIDER

The IPA subdomains provider behaves slightly differently if it is configured explicitly or implicitly.

If the option ´subdomains_provider = ipa´ is found in the domain section of sssd.conf, the IPA subdomains provider is configured explicitly, and all subdomain requests are sent to the IPA server if necessary.

If the option ´subdomains_provider´ is not set in the domain section of sssd.conf but there is the option ´id_provider = ipa´, the IPA subdomains provider is configured implicitly. In this case, if a subdomain request fails and indicates that the server does not support subdomains, i.e. is not configured for trusts, the

IPA subdomains provider is disabled. After an hour or after the IPA provider goes online, the subdomains provider is enabled again.

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.

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.

EXAMPLE

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

```
[domain/example.com]
id_provider = ipa
ipa_server = ipaserver.example.com
ipa_hostname = myhost.example.com
```

SEE ALSO

sssd(8), **sssd.conf**(5), **sssd-ldap**(5), **sssd-krb5**(5), **sssd-simple**(5), **sssd-ipa**(5), **sssd-ad**(5), **sssd-sudo**(5), **sss_cache**(8), **sss_debuglevel**(8), **sss_groupadd**(8), **sss_groupdel**(8), **sss_groupshow**(8), **sss_groupmod**(8), **sss_useradd**(8), **sss_userdel**(8), **sss_usermod**(8), **sss_obfuscate**(8), **sss_seed**(8), **sssd_krb5_locator_plugin**(8), **sss_ssh_authorizedkeys**(8), **sss_ssh_knownhostsproxy**(8), **sssd-ifp**(5), **pam_sss**(8). **sss_rpcidmapd**(5)

AUTHORS

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