This page was last updated on Nov 05, 2025.

Introduction

SSSD is an acronym for *System Security Services Daemon*. It is the client component of centralized identity management solutions such as FreeIPA, 389 Directory Server, Microsoft Active Directory, OpenLDAP and other directory servers. The client serves and caches the information stored in the remote directory server and provides identity, authentication and authorization services to the host machine.

Why do you need SSSD

Did you ever wonder how the same login works on all of the computers in the school lab? That's because they are joined to a domain that's part of an identity management solution. A centralized, redundant, and secure set of servers to manage users, groups, policies and more.

This is not necessary when you have a handful of machines to manage but when scaled up to tens, hundreds or thousands then SSSD becomes an important Linux directory-client component. You certainly can force the new guy to go to each machine to create a user, update their password, remove a user. However, that's pretty mean and a waste of time when there is a much easier way of doing things.

Identity management solutions

The community project FreeIPA is an Identity Management solution. It's also known as the Red Hat IdM or simply IPA. FreeIPA uses 389 Directory Server as its database. Directory servers contain objects like a user object. That user object contains user information stored in LDAP attributes and it is accessible on the network through the LDAP *(Lightweight Directory Access Protocol)* protocol.

Microsoft Windows Server implements a directory service named Microsoft Active Directory Domain Services, abbreviated as Active Directory or AD. Active Directory is another identity management solution that is specifically designed for Windows. It also uses an LDAP server and has many similarities with IPA.

Beside FreeIPA and Active Directory, SSSD can also integrate to other identity solutions using the LDAP provider (for pure LDAP servers) and the Kerberos provider (for Kerberos authentication instead of plain passwords). Additionally, some OAuth2.0/OIDC based Identity Providers (IdPs) are supported via the IdP provider. It can also directly integrate with local users using the proxy provider.

Note

Do not confuse LDAP with 389 Directory Server. 389 Directory Server is one of the projects that implements the LDAP protocol.

See also

Checkout the rest of this website and following manual pages for more information on the different kinds of providers.

- man sssd-ldap
- man sssd-krb5
- man sssd-ipa
- man sssd-ad
- man sssd-idp

For more information about FreeIPA and other compatible directory servers, please check out the following links.

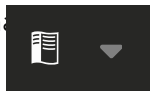
- 389 Directory Server
- FreeIPA
- Microsoft Active Directory
- OpenLDAP

Merging Linux and Windows worlds

IPA and AD can have a trust created between the domains. Why? Consider if there was a company merger or acquisition. There won't be a need to migrate users but instead, set up a trust and allow Bob from industry giant "SSSD Corp" to administer newly acquired small business "LDAP LLC".

It's not uncommon for companies to have two departments, one that supports and administers Linux and the other team manages Windows, so each will have control over its own users and groups. Using SSSD's IPA provider, AD domain users can then log in to a host managed by IPA because of the established trust between domains.

SSSD can authenticate directly to AD and there will be no need for IPA servers. Users and groups will be entirely managed by AD. So why use indirect authentication? IPA is more flexible for managing Linux hosts than AD since it can manage SUDO rules, SELinux users and contexts, automount maps, and more. If a company employs separate Linux and Windows teams, SSSD-AD direct integration may result in Linux teams waiting for AD users to be created. Hopefully, you can start to see th solution has its pros and cons and will not be discussed in this introduction.



Access control

What else is important to know about SSSD? We've discussed authentication, verifying the user and that the correct password is being used. We have not discussed if that user is **allowed** to login, this is what we mean by authorization. Specific users and, or groups can be authorized to access the host using centrally managed HBACs *(Host-Based Access Control)* or filters in the SSSD configuration. For example, web admins don't need access to all the servers on the network and only need access to the Apache and Nginx servers. This level of granular control can easily be done. The `ad_provider` can use AD's GPOs *(Group Policy Objects)* for HBACs.

In addition to access control, SSSD can lookup SUDO rules and roles to control what users can do and when they can do them on the system. For example, a web admin only needs access to stop, start and reboot the httpd service and edit the httpd configuration file. Any failed attempt to escalate their access will also be logged.

Offline access

What happens when if the directory server is offline? Or you have a poor wireless signal on your laptop and keep on disconnecting? Do you need to log in with a local user? Absolutely not, SSSD has a cache to improve performance after finding a user. The user's password is also stored (if enabled) and as long as the user has logged in before, they will be able to log in with SSSD even when it is unable to communicate to the servers.

See also

Do you want to start using SSSD? Check out the Quick Start Guide. You can then follow with SSSD Architecture and introductions to specific components.

Go back: Quick Start Guide

See next: SSSD Architecture