



Technology Transfer Partners

Advancements in Linux Authentication and Authorisation using SSSD

Lawrence Kearney
Enterprise Service and Integration Specialist
TTP Advisory Board Member for Higher Education, Americas

e. lawrence.kearney@earthlink.net
w. www.lawrencekearney.com

Origins in the freeIPA project

(Identity, Policy and Audit)

There is a freeIPA client

Red Hat originates a new client project

Narrower in scope

Provided funding and (2) dedicated developers

Commercially viable software base to bubble up from the
Cent OS and Fedora projects

SSSD package description:

Provides a set of daemons to manage access to remote directories and authentication mechanisms.

Provides an NSS and PAM interface toward the system and a pluggable backend system to connect to multiple different account sources.

It is also the basis to provide client auditing and policy services for projects like FreeIPA.

Thank goodness! A name change opportunity is upon us!

Seriously ?!

“System Security Services Daemon”

We would have very happily accepted:

“Single Sign on Service Daemon”

“Simple Sign on Solution Daemon”

Even:

“Simplesmente Autenticação Serviços Daemon”

What need is SSSD addressing?

PAM and NSS frameworks have scaling caveats, and are becoming legacy as identity management frameworks evolve

Linux servers currently aren't ideal federation platform candidates as a result

LDAP directories are becoming more specialised and are proliferating

Better Active Directory integration is more mission critical

Local files

... ticked, next

Network Information Service (NIS)

... ticked, next

pam_unix nss_ldap

Local authentication, remote user store
Password management
No session management

pam_ldap nss_ldap

Secure remote user lookup and authentication
Password management
No session management

`pam_ldap` `pam_krb5` `nss_ldap`

Secure remote user lookup and authentication

Password management

Session management (SSO capable)

MIT kerberos capable

MS Windows® and Active Directory for Domains capable

`pam_ldap` `pam_krb5` `pam_winbind` `nss_ldap`

Secure remote user lookup and authentication

Password management

Session management (SSO capable)

MIT/MS Windows® kerberos capable

MS Windows® RPC capable

MS Windows® and Active Directory for Domains capable

MS Windows® file share participation

Name Service Caching daemon (nscd)

- Next query caching for users, groups, hosts and services
- No offline authentication but can maintain active sessions

Windows Bind daemon (winbindd)

- Does not require remote posix attributes
- Requires AD Domain joining
- Serves as a front end for PAM, NSS and Samba

LDAP Name Service daemon (nslcd)

- Simplified configuration file
- Requires remote posix attributes
- Does not require AD Domain joining

Large scale deployments become complex

Workforce and administrator skill set considerations

Authentication service enhancements

Greater extensibility

Multiple concurrently available identity stores

ID collision features

SSL/TLS or SASL/GSSAPI is required

Kerberos and SSO features

Reduced server loads

Offline authentication

Configuration consolidation

Backward compatible with legacy PAM / NSS stacks

Legacy PAM / NSS / winbindd¹ modules not required

Integrates with winbindd if necessary

Integrated service configurations (ssh, sudo, autofs etc.)

Single configuration file, reduced complexity

MS Windows® or Samba file shares

Still require winbindd be configured and used

NFS file shares

May still require nscd but without user and group caching

Interactions with some older linux applications

Those that aren't flexible concerning case

Those that will only talk to legacy PAM and NSS modules

Migrating from configurations using id mapping can be more complex

Seriously, if I type:

“SSSH” or “SSSL”

One more time I may scream !!

[sssd] Global parameters

services =

domains =

[nss], [pam], [sudo] Service parameters

reconnection_retries =

filter_users =

[domain/NAME] SSSD domain parameters

id_provider =

auth_provider =

chpass_provider =

access_provider =

SSSD Domain = Identity Provider + Authentication provider

SSSD uses a parent/child process monitoring model

[sssd] Parent process, **Monitor**

[nss] Child process, **Responder**

[domain/LDAP] Child process, **Provider**

SSSD process example:

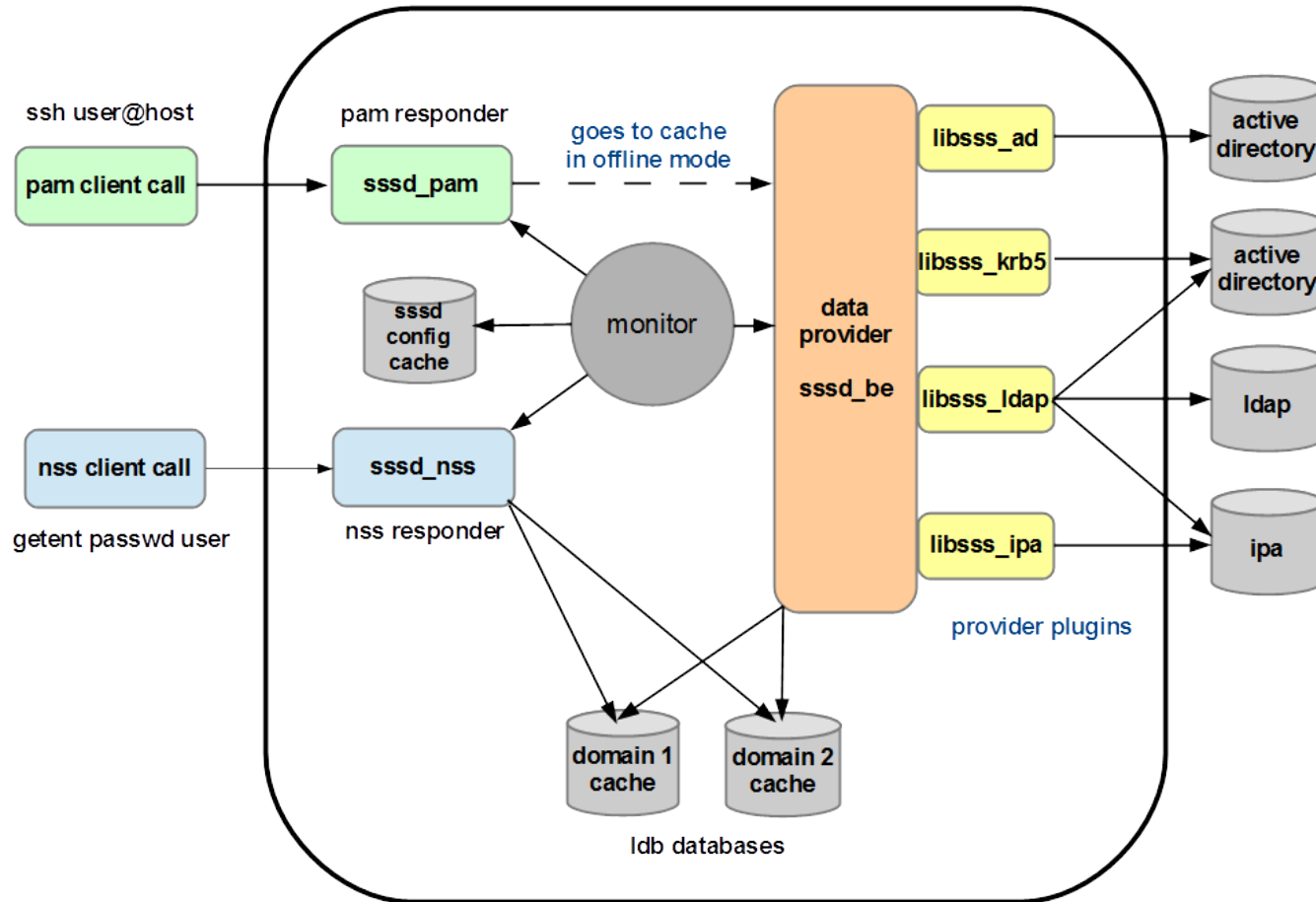
```
ps -eaf | grep sssd
```

```
root      1476      1          0   /usr/sbin/sss
root      1478     1476        0   /usr/libexec/sss
root      41279    1476        0   /usr/libexec/sss --domain LDAP
```

```
pstree -A -p 1476
```

```
sss (1476)  - + - sss_be (41279)
              | - sss_nss (1478)
```

“SSSD” architecture overview



Local	Accounts are kept in a local ldb database
LDAP	Relies on installed extensions of target directory
Kerberos	Relies on installed extensions of target directory
AD	Supports many native Active Directory features
iPA	Supports trusts with Active Directory domains
IdM	Integrates tightly with RHEL IdM implementations
Proxy	Permits integration of other provider modules

Id, Authentication, Access and Changing Passwords

`id_provider = ldap, ipa, krb5, ad, proxy`

`auth_provider = ldap, ipa, krb5, ad, proxy`

`access_provider = permit, deny, ldap, ipa, ad, simple`

`chpass_provider = ldap, ipa, krb5, ad, proxy, none`

- Most providers fulfill multiple roles
- Different providers can, and often are be combined

Local

- Enhanced local account features
- Familiar local user management tools

LDAP

- Flexible attribute mapping capabilities

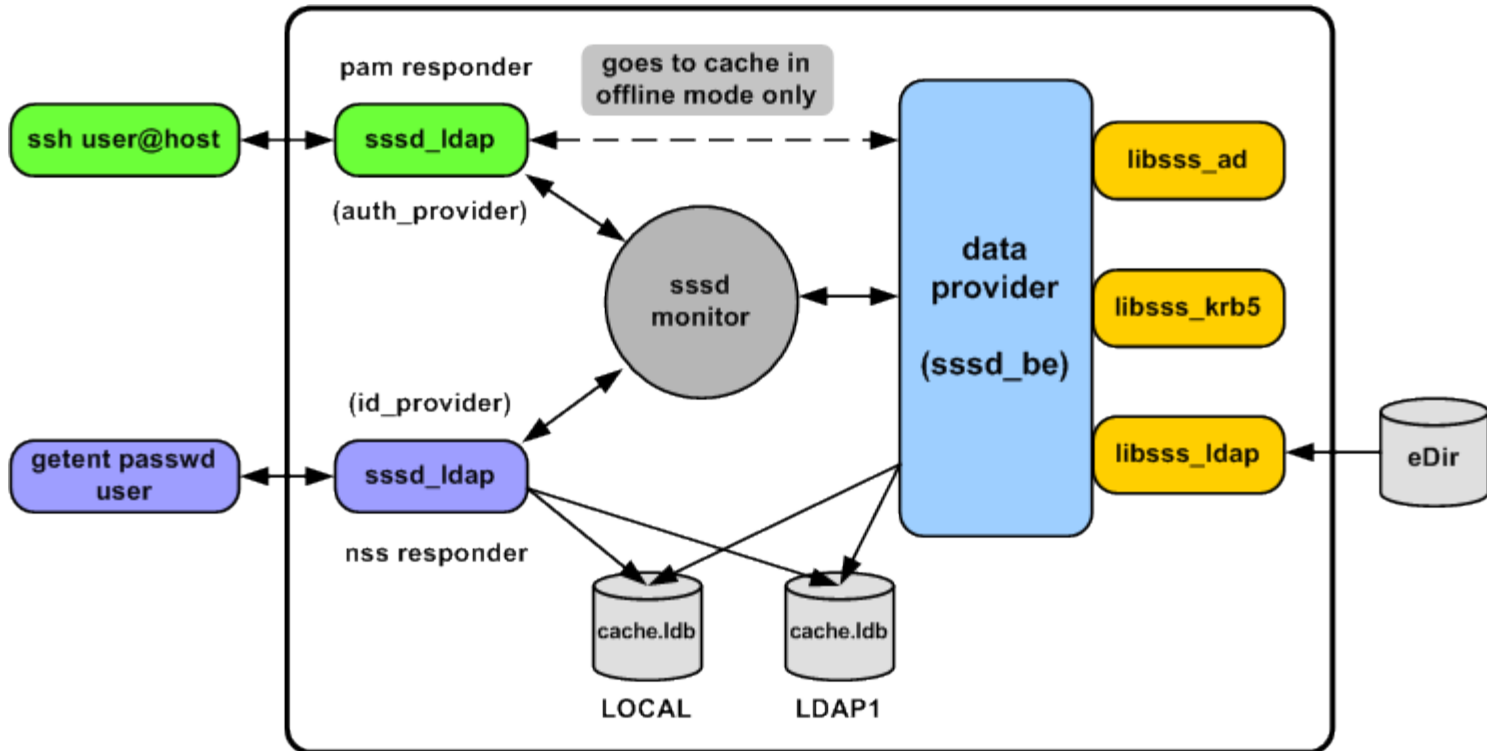
Kerberos

- SASL/GSSAPI support improves application support

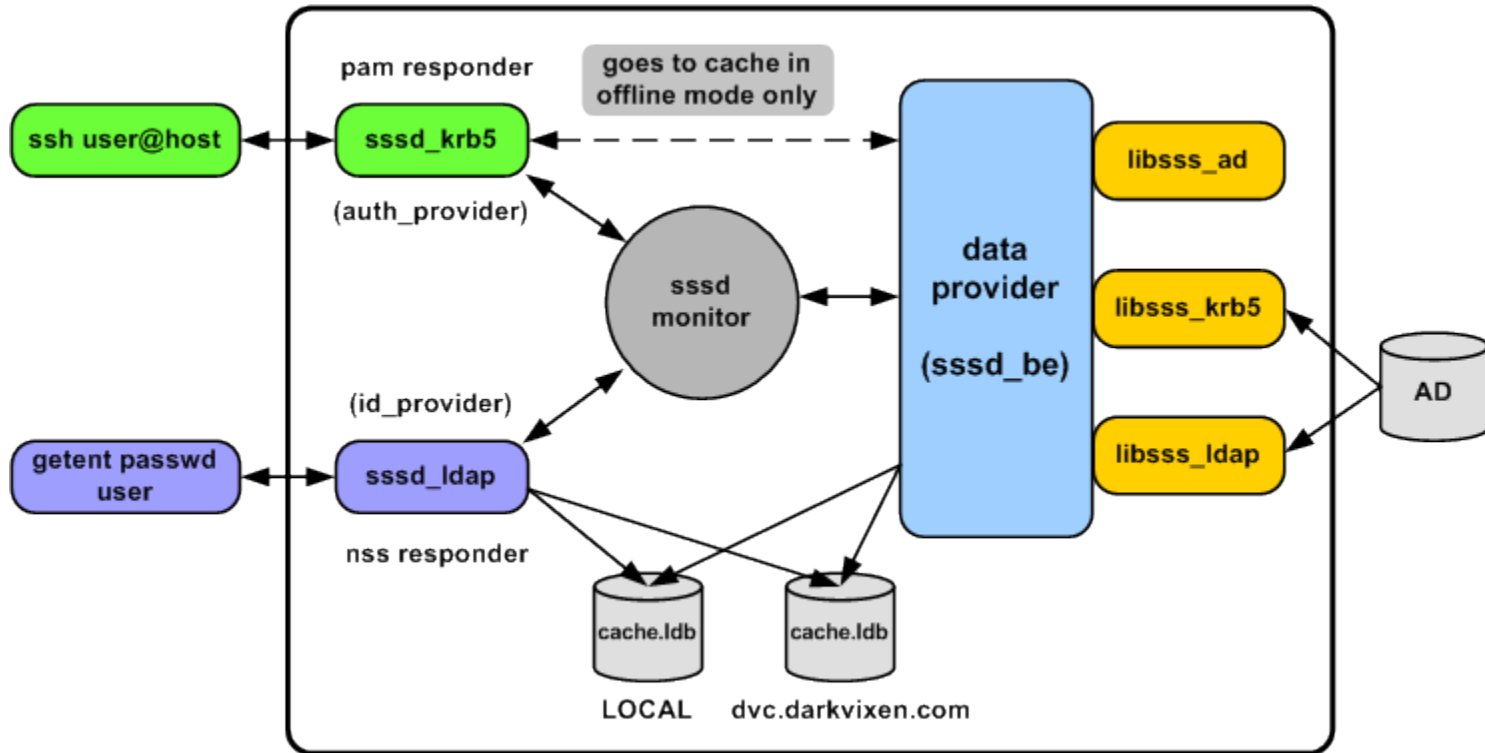
AD

- Login performance improvements
- Trust and domain auto-discovery features
- Native schema, DNS update and security support

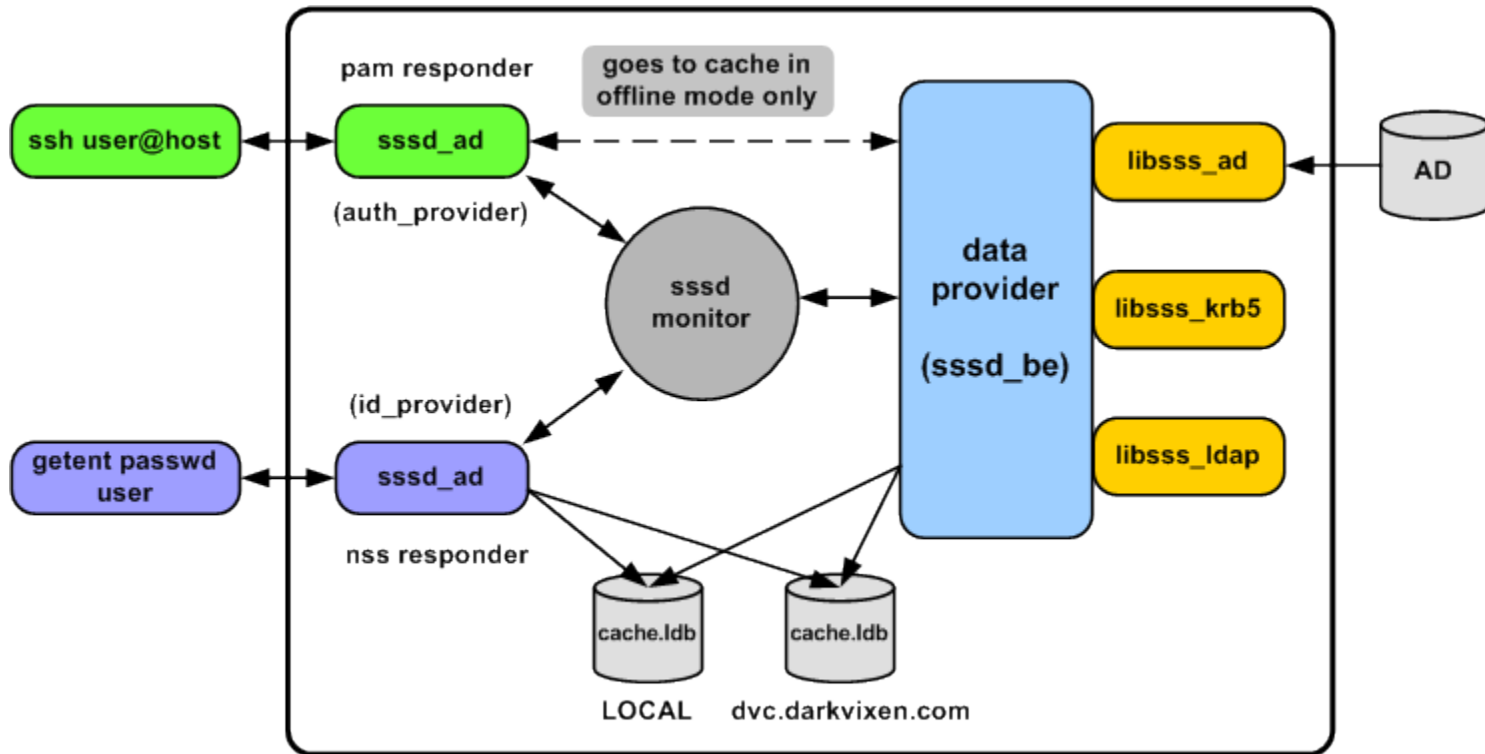
SSSD identity provider example



SSSD identity provider example



SSSD identity provider example



Many linux distributions are now SSSD aware

Auto-configuration using native distribution utilities

Enterprise linux distributions include

Red Hat Enterprise Linux 5.6:	SSSD 1.5
Red Hat Enterprise Linux 6:	SSSD 1.9
Red Hat Enterprise Linux 7:	SSSD 1.11
Suse Linux Enterprise Server 11.2:	SSSD 1.9
Suse Linux Enterprise Server 12:	SSSD 1.11

Identify existing services that should be modified

PAM LDAP and NSS LDAP configurations
NSCD user, group, host or service caching

Determine how posix attributes will be provided

Provided by directory service or linux ID mapping

Install software on your platform

Typically samba and kerberos are required for initial setup²
Not all distributions package the SSSD similarly

Configure transport security

TLS/SSL for eDir over LDAP

TLS/SSL for AD over LDAP

SASL/GSSAPI for AD over LDAP/kerberos

Configure SSSD identity providers and access control

Identity and access control providers can be mixed

Suse and Red Hat are aligning with AD integration needs

Would like to see the AD id provider included in SLES 11.3

SSSD 1.11

Red Hat: Realmd utility will auto-configure AD id provider

Suse: YaST Authentication client will auto-configure AD id provider

Expanded AD access control provider

NetBIOS/DNS domain name auto-discovery

Developing with 1.11

AD access control provider will include group policy support

SSSD CIFS integration



Thank You !

Lawrence Kearney

e. lawrence.kearney@earthlink.net

w. www.lawrencekearney.com