# Authenticating with SSSD

**Don Vosburg**
Systems Engineer, SUSE
dvosburg@suse.com

**Rodolfo Bejarano**
Systems Engineer, SUSE
rbejarano@suse.com

# How do you authenticate?

# Agenda

What is SSSD?

SSSD Advantages

Why should I care?

Configuration with multiple authentication sources

ACL and authorization

Demonstration

Questions and Answers

# What is SSSD?

SSSD (System Security Services Daemon) is a system daemon whose primary function is to provide access to identity and authentication remote resource through a common framework that can provide caching and offline support to the system.

It provides PAM and NSS modules, and provides a better database to store local users as well as extended user data.

# SSSD Advantages

Authentication service enhancements

- Greater extensibility
- Multiple concurrently available identity stores
- ID collision management features
- SSL/TLS or SASL/GSSAPI is required
- Single configuration file
- Reduced server loads
- Offline authentication

# SSSD Providers

identity, authentication, password, autofs, sudo, etc

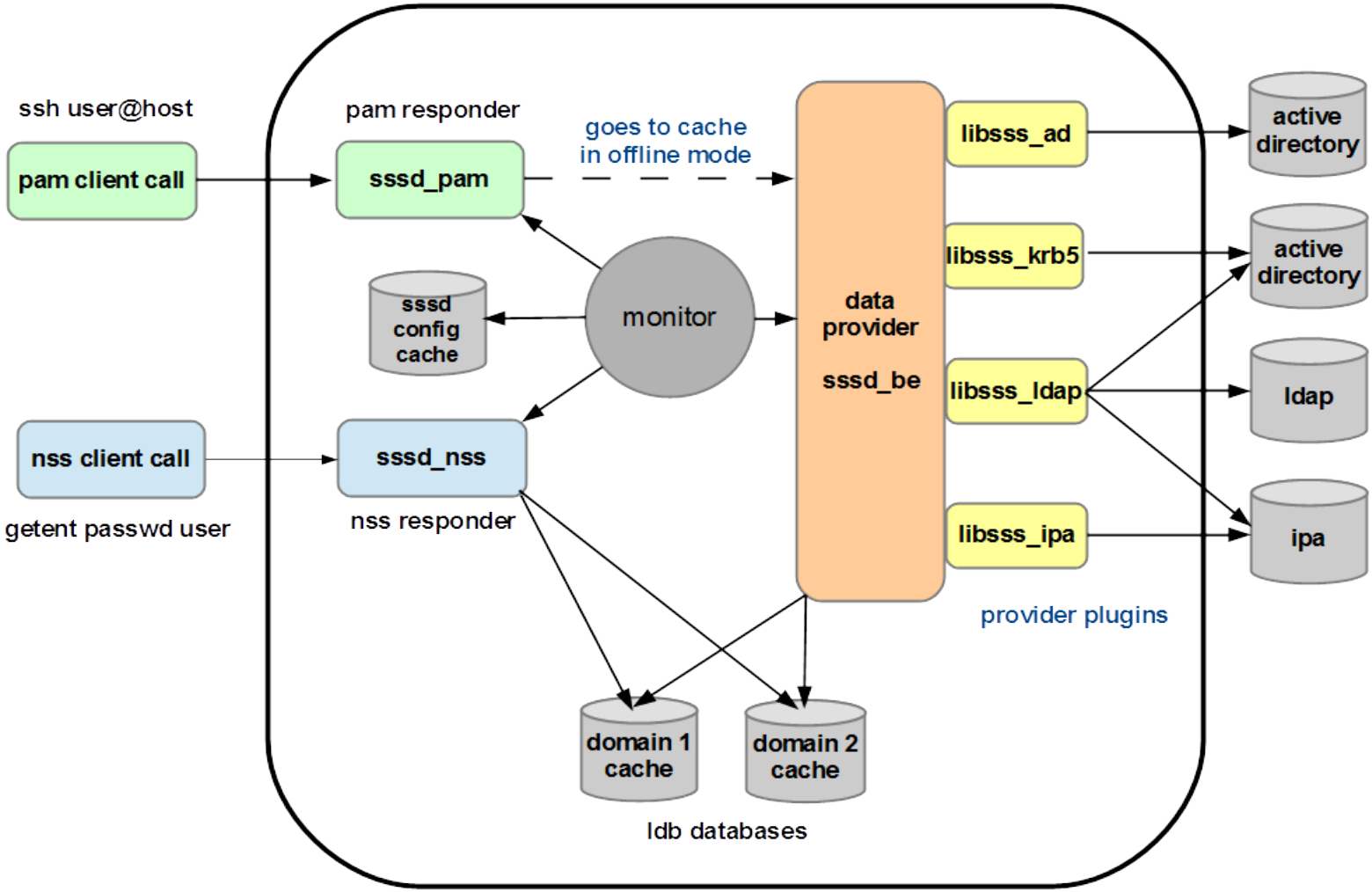
| | |
|---|---|
| Local | Accounts are kept in a local database |
| LDAP | Relies on installed extensions of target directory |
| Kerberos | Relies on installed extensions of target directory |
| AD | Supports many native Active Directory® features |
| iPA | Supports trusts with Active Directory® domains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

# SSSD Processes

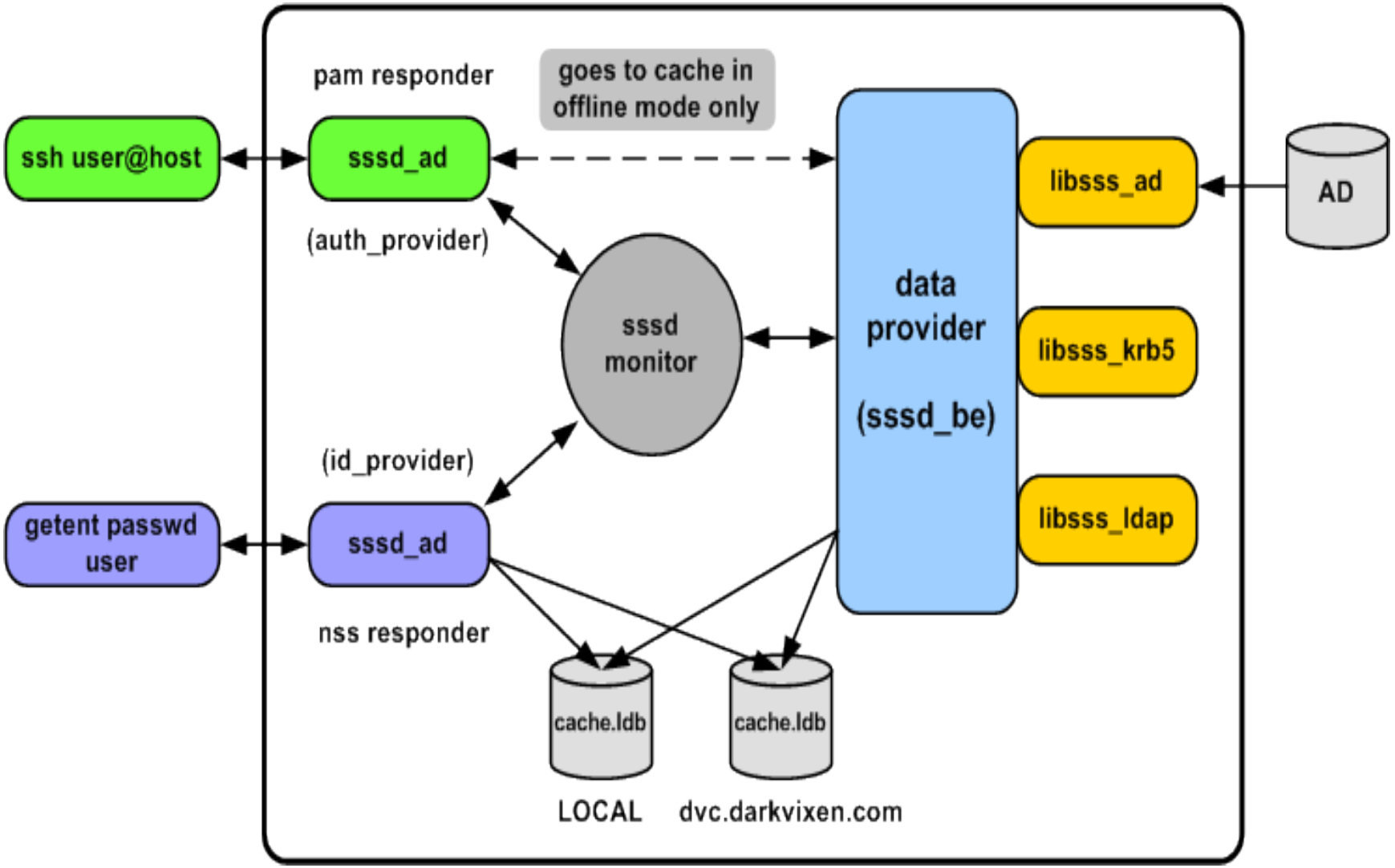
SSSD uses a parent/child process monitoring model

| | |
|---|---|
| [sssd] | Parent process, Monitor |
| [nss] | Child process, Responder |
| [domain/ad.dom] | Child process, Provider |

# SSSD Architecture



# Active Directory ID and Auth Providers



# Why should I care about SSSD? (Part 1)

- Replaces less desirable authentication solutions
  - winbind
  - nscd
  - pam_ldap
  - nss_ldap
- Secure framework for multiple authentication domains
- Separately called/controlled Kerberos tickets
- Excellent way to connect to AD
- Integrated with LDAP client in YaST in SLES 11SP3+

# Why should I care about SSSD? (Part 2)

- ACL's are an integral part
- Active upstream development
- Included/supported in

  - SUSE Linux Enterprise Server 11 SP3+
  - SUSE Linux Enterprise Server 12
  - CentOS/Red Hat Enterprise Linux 6/7

# Clash of the Cache – sssd vs nscd

- Historically **nscd** has been used to cache credentials
- Disable it if you do not need host caching
- Can/should still be used to cache hosts/services only
- Confirm that in /etc/nscd.conf

```
enable-cache    passwd    no
enable-cache    group     no
```

# How is SSSD set up?

- Required packages:
  - sssd, krb5_client
- Configure LDAP or Authentication Client in YaST
  - This will configure nsswitch.conf and pam settings
  - If you do not need LDAP, you can use it as a way to discover proper settings
- Optionally manually configure krb5.conf, sssd.conf, nsswitch.conf, and the common stack in /etc/pam.d

# Configuring sssd.conf - p1

```
[sssd]
config_file_version = 2
services = nss,pam
domains = default,AD
# SSSD will not start if you do not configure any domains.
# Add new domain configurations as [domain/<NAME>] sections,
# then add the list of domains (in the order you want them to
# be queried) to the "domains" attribute comma delimited.

[nss]
filter_groups = root
filter_users = root
[pam]
```

# Configuring LDAP in sssd.conf (p2)

```
[domain/default]
ldap_uri = ldap://sssd-demo.addomain12.com
ldap_search_base = dc=openldap,dc=addomain12,dc=com
ldap_schema = rfc2307bis
id_provider = ldap
ldap_user_uuid = entryuuid
ldap_group_uuid = entryuuid
ldap_id_use_start_tls = True
enumerate = True
cache_credentials = True
ldap_user_search_base = dc=openldap,dc=addomain12,dc=com
ldap_group_search_base = dc=openldap,dc=addomain12,dc=com
chpass_provider = ldap
auth_provider = ldap
```

# Configuring AD – AD Provider sssd.conf (p3)

## AD using the AD Provider (requires sssd v 1.02+)

```
[domain/AD]
 id_provider = ad
auth_provider = ad
access_provider = ad
ad_server = server.ad.example.com
default_shell = /bin/bash
fallback_homedir = /home/%d/%u
use_fully_qualified_names = True
```

# Configuring AD with LDAP (p4)

**AD using LDAP and NIS - traditional method**

```
[domain/AD]
 id_provider = ldap
 auth_provider = krb5
 chpass_provider = krb5

 ldap_uri = ldap://win2012r2.addomain12.com
 ldap_search_base = dc=addomain12,dc=com
 ldap_schema = rfc2307bis
 ldap_sasl_mech = GSSAPI
 ldap_user_object_class = user
 ldap_group_object_class = group
 ldap_user_home_directory = unixHomeDirectory
 ldap_user_principal = userPrincipalName
 ldap_account_expire_policy = ad
 ldap_force_upper_case_realm = true
 ldap_referrals = false
 cache_credentials = True
 krb5_server = win2012r2.addomain12.com
 krb5_realm = ADDOMAIN12.COM
```

# Using ACL's with sssd.conf

Two ACL providers:

1 - **Simple Access Provider**

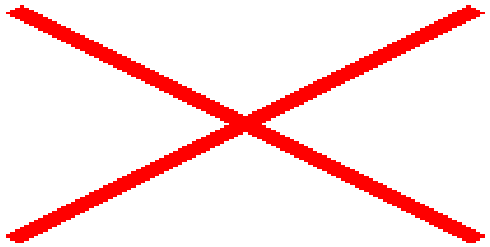
- Granting or denying access for objects is based on a content of “allow” and “deny” lists

2 - **Active Directory Access Provider**

- New access filter option to AD access provider
- More advanced format can be used to restrict the filter to a specific domain or a specific forest

# Simple ACL example

- Setting environment:



- Example SSSD rules:

- Results:

– Granted access to: jerry and spike

– Denied access to: tom (denying rules take precedence over accepting rules)

# Reference material

- SSSD website

  https://fedorahosted.org/sssd/

- Configuring AD using AD Provider

  https://fedorahosted.org/sssd/wiki/Configuring_sssd_with_ad_server

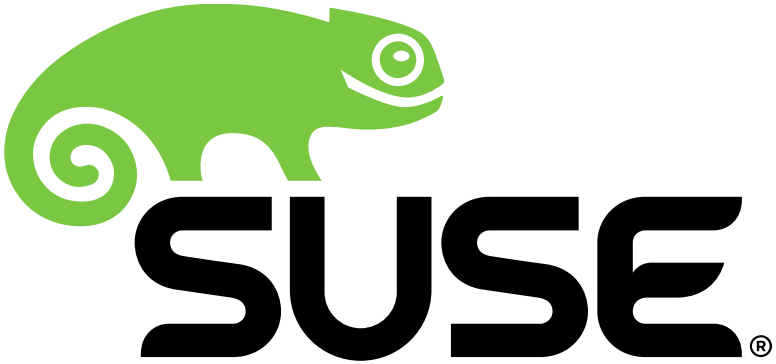
- ACL presentation by Pavl Reichl

  http://www.freeipa.org/images/c/cc/FreeIPA33-sssd-access-control.pdf

# Demonstration

# Your Questions!

Thank you.


SUSE®
We adapt. You succeed.