redhat.

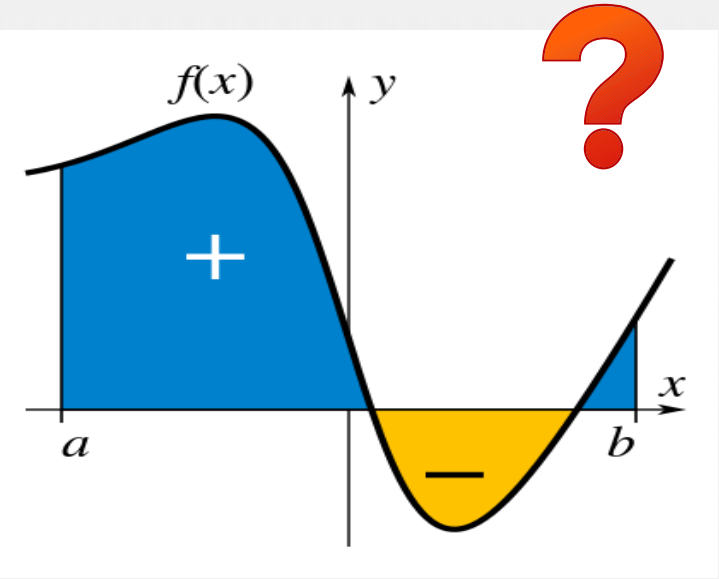
Samba and Linux Distributions

Let's integrate better

Simo Sorce
Sr. Pr. Sw. Engineer - Identity Management and Security

Samba XP 2016
2016-05-11

Integration ?

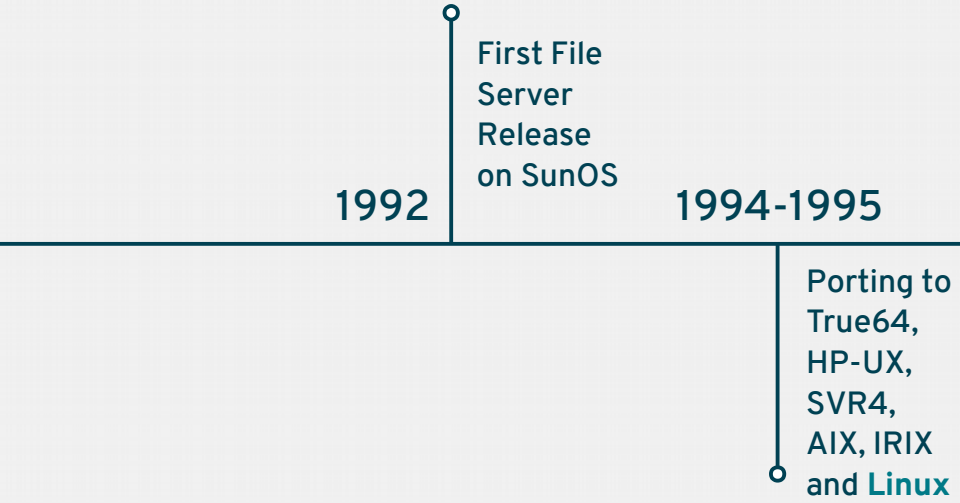


In engineering, system integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a system.

https://en.wikipedia.org/wiki/System_integration

A brief Samba history

In the beginning ... there was a file server



Samba is just a File Server

The extent to which it integrates with the OS is its configuration file. Any more integration is hard due to the compatibility layer to abstract away the specifics of each of the UNIX flavors and their file system.

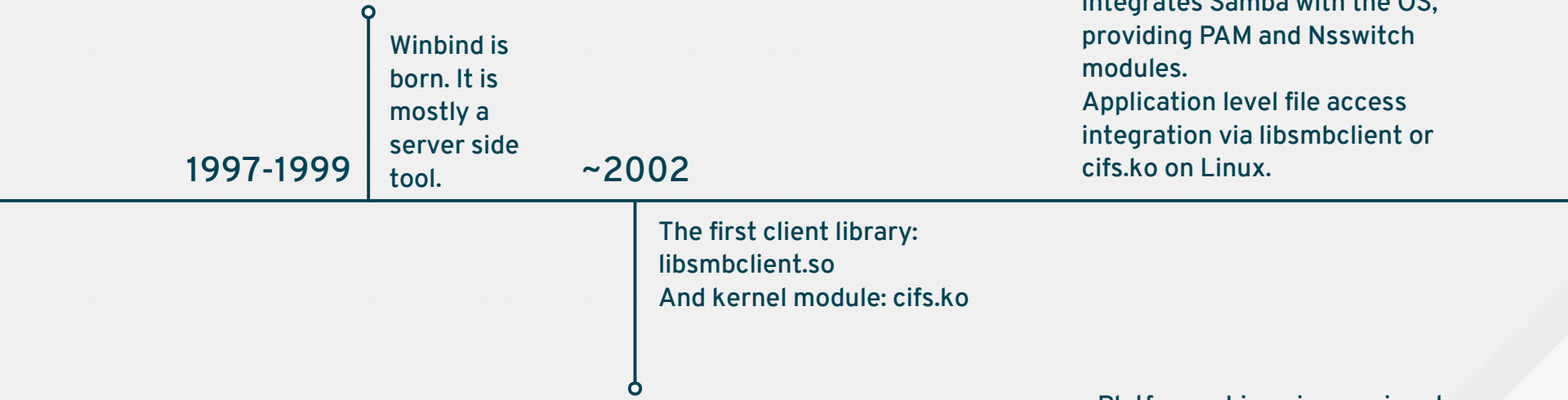
Platforms: any UNIX flavor you can remember

A brief Samba history

More components arise

Samba is still mostly a File Server (can act as NT4 Domain Controller)

Winbind is the first component that integrates Samba with the OS, providing PAM and Nsswitch modules.
Application level file access integration via libsmbclient or cifs.ko on Linux.



Platforms: Linux is prominent, Still many UNIX flavors

A brief Samba history

The modern era

Samba is a full blown AD domain controller as well. Many new protocols and libraries available. Lots of integration “possible”

2005-2007

Novell desktop integration project CODE10

2007-2009

Samba AD was born. Many client libraries available: talloc, tevent, tdb, ldb, dce/rpc

2011-2016

FreeIPA uses Samba for AD Forest Trust support

Platforms: Linux
Other platforms partially supported

Samba and Linux/UNIX distributions

- On classic UNIX Samba has always been an add-on component
  - Generally distributed via some “Free Software” side channel, only File Server
- With the advent of Free Software based operating systems Samba has become a first class citizen of the OS, usually available in the main distribution channel
  - Samba is the default (and generally only) Windows compatible File Server solution
  - Samba has historically been the Windows Domains integration tool both on File Servers and Workstations
  - Many distributions now provide Samba as an AD-compatible Domain Controller
- Samba is also available in many embedded systems and appliances, from small “personal” devices to big “enterprise” clustered/distributed file systems

All good! We are done!

- Questions ?

Integration with the OS

On the server

User identities and Identity Mapping for Access Control purposes (domain member or controller)

Serve files bridging local filesystem (using as many features as possible) to NTFS semantics

Eanble Krb5/NTLM SSO for some server applications (squid, etc..)

On the desktop

User Authentication and Identity

Credential caches to enable user agents to SSO into services (browser, email, files, etc..)

Access to file servers

GPO settings for machines/users

How is it working ?

We are doing ...

- ... ok on the server
- ... so-so on the desktop

Key integration points

- Support and integration with the distribution libraries and APIs
  - Distro preferred ccache types, Krb5 and GSSAPI libraries
- Enablement of client applications and desktop environments
  - Browsers and file managers need to be able to seamlessly work with all credential caches and file server's protocols to deliver a good user experience
- Standard login component that properly initializes and maintain credential caches
  - Winbind and SSSD play this role in modern distributions

- and finally -

- MAKE IT SIMPLE !

1. Security libriaries and Credential Caches

- Credentials Caches are useful only when all the OS components can properly use them
  - Common libraries used throughout the system
  - Support for the system-preferred credential cache format
- Security Libraries:
  - Use a standardize and common APIs (GSSAPI)
  - Reduce attack surface by standardizing on a smaller set of libraries (especially when crypto is involved)
- Bring features to the system libraries so that the whole system benefits at once
  - Reduce confusion when different components support different things due to multiple implementations being used at the same time in different parts of the system

Security Libraries: Krb5

- Fedora and RHEL standardized on MIT Kerberos for historical reasons
- Samba used to be krb5 library agnostic
  - When it started using Heimdal went as far as removing support for MIT Kerberos
- The problem is that you cannot mix MIT Kerberos libraries and Heimdal libraries in the same applications
  - Although libraries can be segregated to some degree through Linker options, there are many side effects and mismatched support for features to account for as well
  - Example: Fedora and RHEL default to Keyring ccaches which Heimdal does not support

Security Libraries: GSSAPI

- GSSAPI is a well defined and standardized (by IETF) API and Network protocol
  - Compatible at the networking level with Microsoft Windows' SSPI
  - Supports multiple mechanisms like Krb5 and NTLMSSP (and others, like GSS-EAP)
- Many improvements have been made to GSSAPI and SPNEGO support upstream
- A special feature called “interposition” has been added in recent years to allow privilege separation at the system level which may allow:
  - Secure use of system credentials (host keytab) on Linux systems using with multiple aliases for the same key as Windows does by default, maintaining separation of privileges between services (no direct access to keying material). Implemented today in the GSS-Proxy project.
  - Potentially allows secure use of cached NTLM credentials by keeping the “clear-text-equivalent” NT hash out of the hands of user processes.
  - Works only if GSSAPI is performed entirely via the MIT Kerberos mechglue

Security Libraries: NTLMSSP

- GSSAPI supports the use of the NTLMSSP mechanism
  - both RAW and encapsulated in SPNEGO
- GSS-NTLMSSP is a project I started to provide access to NTLM based secured channels to more than just the Samba ecosystem
  - Uses GSSAPI a well defined security API that applications can use relatively easily instead of rolling their own broken NTLM support
    - No application (including Samba) implemented NTLM fully and completely.
    - Most have very poor implementations
  - Meant to do one thing only and do it well, it supports the full MS-NLMP spec, including channel bindings and all the additional security features of the latest servers
  - Can use Winbind to allow services to perform domain logons
  - Requires still some usability work to handle cached credentials

Security Libraries: Credential Caches

- Why Fedora/RHEL use this “special” Keyring ccache type anyway ?
  - TGTs are locked in kernel memory and never touch the disk
  - Supports cache collections, meaning a user can keep multiple TGTs available at the same time
  - Solves a chicken/egg problem with user private directories not being available before login completes, in order to avoid storing caches in a system-wide /tmp
- Keyring ccaches are not perfect, and we are considering a further move to a daemon based approach similar to the one used by Mac OS
- Windows systems have always had caches for NTLM credentials, we still lack any reasonable support for that at the system level (Winbind has some support but hard to use).
  - Although NTLMSSP should go the way of the dodo it is still used a lot in Windows-centric networks, we'd like to offer a better experience to those users as well

Security libraries: integration

- Good progress in making all of Samba to build using MIT Kerberos
  - For years we (Red Hat Samba team) have been working to make Samba build again against MIT Kerberos, starting with client libraries
  - Finally the AD DC required parts are close to be ready for production use
- Additional work:
  - Complete transition from Heimdal to MIT Kerberos (Domain Controller case)
  - Fix any remaining bugs in the system GSSAPI/SPNEGO library
- Possible future working items:
  - Make GENSEC a shim layer above the standard system GSSAPI
  - Use GSS-NTLMSSP within Samba

2. Application Enablement

- We've made great progress with the plumbing, it's time to work on the porcelain
- The desktop experience is still far from ideal and requires **way too much** customization by the user
- A lot of new applications keep coming out without support for “enterprise” login features
  - No GSSAPI, Krb5 or NTLMSSP support even when standard extesions are available for the implemented protocols
- Browsers are still far from offering a good experience on the Linux desktop
  - Please attend Alexander's talk, he'll go all over it!
- The experience in general hit-and-miss
  - We need to standardize on a single interface to make developers' life reasonable
  - Testing is a big issue

3. Common login daemon

- Samba provides Winbind
- However Fedora and RHEL have been standardizing on SSSD for a while.

- How do we seamlessly integrate login services in this situation ?

- For the current integration story see Sumit's talk about SSSD and Winbind

Why SSSD anyway ?

- It is a project focused on the general login and network identity problem, not specific to Windows Domains (Originates in the FreeIPA project)
  - LDAP, NIS, Krb5, etc… support proxying to legacy PAM/NSS modules as well
- Provides various additional security services to the system beyond the trditional PAM/nsswitch modules
  - Client libraries enabled by default in the OS
  - No code runs in applications and no restart is necessary when configuration changes are made
  - Multiple configuration files (PAM, nsswitch, sudo, etc) now need not be touched, a single place deliver all data and configuration for a coherent security configuration of the system
- We are strongly considering making it handle local users (in a backwards compatible way) as well one day
  - Support for additional features for local users as well in a coherent way across the system

SSSD and Winbind

Feature	Winbind	SSSD
Complex AD setups	Y	(Y)
Other IdM Servers	N	Y
CTDB Support	Y	N
NTLM Auth	Y	N
Offline Mode	Y	Y
Fast nsswitch cache	N	Y
SSH Keys, Sudo Rules, HBAC, SeLinux user-role, autofs maps	N	Y
DBUS Interface	N	Y
GPO support	(N)	(Y)

Can we integrate SSSD and Winbind ?

- Why do we need integration ?
  - To make user's life easier and make it “just work”(TM)

- Winbind currently “owns” the Domain level NTLM logon space
  - Hard to rebuild from scratch due to SMB/RPC infrastructure needed to implement it
  - Restriction to a single schannel client per host requires collaboration between tools
- Winbind is compatible with CTDB clusters

- Can we isolate these features in a “Winbind backend” component (LSAD ?) that can be used by SSSD or the “Winbind frontend” seamlessly ?
- Is there another solution that allows deper integration and not just coexistence ?

Make it simple!

- We need to look beyond the technical details
- What can we do as a community to make it simpler to run Samba ?
  - As a simple file server
  - As a domain controller
  - On the client
- What can we do to make it easier to use Samba services by other projects ?
  - Libraries
  - Protocols
  - CLIs

We are all in the same Team

(Really, we are)

Let's work together to make users happy!

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos