Progress with passwordless Fedora for enterprise and standalone use

Alexander Bokovoy

Sr. Principal Software Engineer // Red Hat

Passwordless Fedora

Replace passwords with better alternatives

Passwordless does not necessary mean without a password

Focus is on enterprise systems

- Systems organized by enrolling them into organizational units (domains)
- Systems controlled by centralized decision making
  - cost of individual maintenance of authentication details is high
- Authenticated context does matter when applied across multiple systems
  - Authentication happens too many times to get in the way of doing a business

Best practices would need to apply to individual systems as well

FreeIPA

Passwordless options since 2010

- [2010] Smartcards
- [2013] TOTP/HOTP tokens with a pin (pin = user password)
- [2013] RADIUS proxy (whole credential is sent to a RADIUS server for verification)
- [2021] External OAuth2 identity provider integration
  - No credential stored or requested, OAuth2 device authorization grant is requested instead
- [2023] FIDO2 passkey
  - FIDO2 usb or NFC-enabled keys

All options result in producing a Kerberos ticket valid for consumption by all Kerberos-aware applications

Kerberos context

Kerberos ticket

- Cryptographically signed data buffers produced by KDC
- Valid for a certain time period and can be renewed for longer
  - FreeIPA issued tickets valid for 24hr by default, renewable for 1 week. Admins can change these policies
- Privilege attribute certificate structure (PAC)
  - Attestation of the client requested the ticket
  - Attestation of the service to which ticket is issued
  - Details of what KDC knows about this client, including membership in the domain groups
- May include information about the method used to obtain the initial Kerberos ticket
  - “Authentication indicators”

All Kerberos-aware applications have ability to inspect this data but cannot not change it

Kerberos context

Practical use: secure and seamless authentication to networking file systems

- NFS and SMB servers do support GSSAPI authentication with Kerberos mechanism
- Both per-user Kerberos credentials and machine account from a system keytab are supported

Kerberos context

Practical use: secure and seamless authentication to remote services

- SSH
  - OpenSSH already has support for GSS API authentication with Kerberos mechanism
  - Work in progress: support authentication indicators in OpenSSH
    - https://github.com/openssh/openssh-portable/pull/500
    - Agreed with Fedora and RHEL maintainers to get it added to Fedora/RHEL soon
- Demo

Kerberos context

Practical use: pam_sss_gss PAM module

- Allows to authenticate a PAM service access with a valid Kerberos ticket
- Requests a ticket to the host service (host/machine.name) using Kerberos ticket in the user's credentials cache
- Capable of checking the method the initial Kerberos ticket was obtained
  - "Allow SUDO access only to users who authenticated with a smartcard"
- Demo

That demo used a KDC on 127.88.88.88?

Yes, running a KDC locally

- An experiment: https://github.com/abbra/local-kdc
  - Configuration to enable KDC to only listen on a loopback interface
  - Uses certmonger to manage locally-issued PKINIT certificates
  - Uses SSSD with proxy id provider and krb5 authentication provider
    - A user defined locally
    - Authentication credentials defined in KDC
  - SUDO uses pam_sss_gss to authenticate using Kerberos tickets

Local KDC? Why?

The evolution of Windows authentication

- Microsoft, October 2023:
  - https://techcommunity.microsoft.com/t5/windows-it-pro-blog/the-evolution-of-windows-authentication/ba-p/3926848

> Our end goal is eliminating the need to use NTLM at all to help improve the security bar of authentication for all Windows users.

Kerberos, better than ever

For Windows 11, we are introducing two major features to Kerberos to expand when it can be used—addressing two of the biggest reasons why Kerberos falls back to NTLM today. The first, IAKerb, allows clients to authenticate with Kerberos in more diverse network topologies. The second, a local KDC for Kerberos, adds Kerberos support to local accounts.

- Microsoft, March 2024: "The Evolution of Windows Authentication" presentation

So why mimic Microsoft?

It is for interoperability and more

- Microsoft realized there is no way to get around Kerberos
  - Already provides secure way of authenticating in a hostile environment (Internet)
  - KDC support already tested well enough by large customers for decades
  - They are not making Active Directory to run on every Windows machine, no
    - Only a KDC to handle Kerberos protocol abstraction behind the scenes
    - Proxy access to it via SMB protocol negotiation with IAKerb Kerberos extension
- We have to handle interoperability demands
  - But we can do much more within the same framework
- Microsoft is not the only one using local KDC
  - macOS has had local KDC in place for ~15 years but decided to retire it
    - The code is still there, one can tweak and enable it like https://github.molgen.mpg.de/pages/bs/macOSnotes/mac/mac_server_kdc.html
  - macOS handled IAKerb for 'Back to My Mack' feature (RFC 6281)

Local KDC

Re-use FreeIPA and SSSD experience and code

- Local KDC means “everything Kerberos”
  - Use the same code for enterprise and local accounts
  - Uniform authentication experience
    - Authenticate with OTP, FIDO2, external OAuth2, etc. in a predictable way
    - Make UI predictable and friendly
  - On-demand peer-to-peer trust mesh
    - Allow machines trust each other without downgrade of authentication for all protocols
- Microsoft is focused on solving NTLM authentication problem only
  - There is no multi-factor authentication support for local KDC in Windows 11 (yet)
  - We can do better security and UX

Work in progress

MIT Kerberos

Support for local KDC

- Allow running KDC over UNIX domain sockets
  - Work in progress: https://github.com/krb5/krb5/pull/1359
- IAKerb protocol extension
  - Current spec draft support merged upstream
    - Not tested against Windows yet as there is no public build of Windows + local KDC/IAKerb available
  - Work has started on auto-discovery

MIT Kerberos

Support for local KDC

- Future work
  - Partial database backend integration with https://systemd.io/USER_GROUP_API/
  - Common MS-PAC issuer plugin for Samba, local KDC, and FreeIPA

Samba

Support for local KDC and IAKerb

- Samba 4.21.0rc1 works already against local KDC if there is any
  - Enables use of local accounts with Kerberos authentication when KDC issues minimal PAC records (which have no LOGON_INFO records)
- IAKerb support in progress: https://git.samba.org/?p=asn/samba.git;a=log;h=refs/heads/asn-iakerb

FreeIPA and SSSD

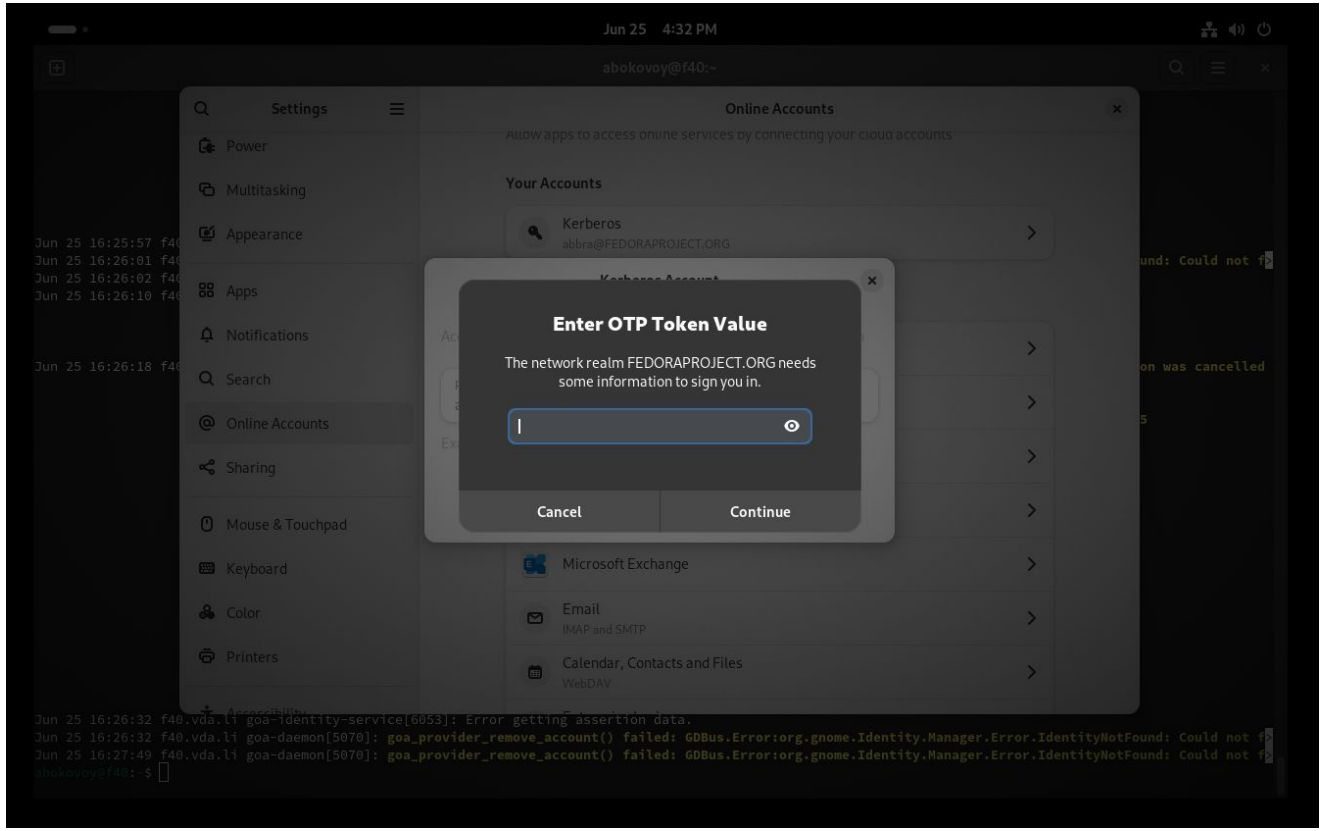
Support for local KDC

- WIP: Allow use of ipa-otpd backend by the local KDC deployment
  - Enables passwordless authentication locally
- IAKerb integration is pending Samba support
- Passwordless auth in PAM and GNOME
  - WIP: New mechanism to negotiate auth methods to PAM applications via JSON extensions in pam_sss
  - GNOME support for passwordless authentication

GNOME support

Passwordless authentication methods

- WIP: redesign of GDM login flow
  - Enables passwordless authentication methods, including QR code and browser integration
- WIP: passwordless Kerberos methods [support in GNOME Online Accounts]
- WIP: various improvements to eliminate asking for passwords where not needed



Thanks!