A local authentication hub

FOSDEM 2025

Alexander Bokovoy & Andreas Schneider

Senior & Principal Software Engineer | Red Hat | Samba Team

Who are we?

About Alexander

- FreeIPA core developer
- Samba Core Team member since 2003
- MIT Kerberos contributor

About Andreas

- Samba maintainer at Red Hat
- Samba Core Team member since 2010
- MIT Kerberos contributor

1

A local authentication hub

Quick overview

Local authentication hub (LocalKDC)

- MIT Kerberos Key Distribution Center (KDC)
- Only accessible locally (Unix domain socket)
- Only runs on demand (Socket activated)

Why do we want a LocalKDC?

- We don't want to enter username/password combos all the time.
- Many applications already have Kerberos and GSSAPI support!

Demo time (#1)

2

POSIX (the old days)

Local POSIX accounts

- Local POSIX accounts are normally defined in /etc/passwd and are password based
- Services like OpenSSH make use of it through the PAM stack
  - SSH keys improve security and simplicity
- Works well for home users
- Doesn't work well for large enterprises

Centralized Identity Management

- Managing thousands of accounts is tough
  - Centralized management is a must
- RHEL IdM / FreeIPA
- Windows Active Directory (AD) or Samba AD

Centralized Authentication

- Users log into different services during the day
- Single Sign-On for all systems
- Kerberos for system level authentication (and OAuth2 for web apps)

On standalone Systems

- Users log into a single system
- Don't usually expect single sign-on to work across the different systems
- **Can we improve by reusing a centralized approach?**

NTLM CRAP

NTLM is (a) CRAP

- New Technology Lan Manager is Microsoft's
- Challenge Response Authentication Protocol

The Battle Of (Lost) Causes

- NT LAN Manager (NTLM, MS-NLMP) is a family of security protocols
  - a handy tool for hacking into your environment
  - more than 30 years of experience
    - first revisions date back to 1993

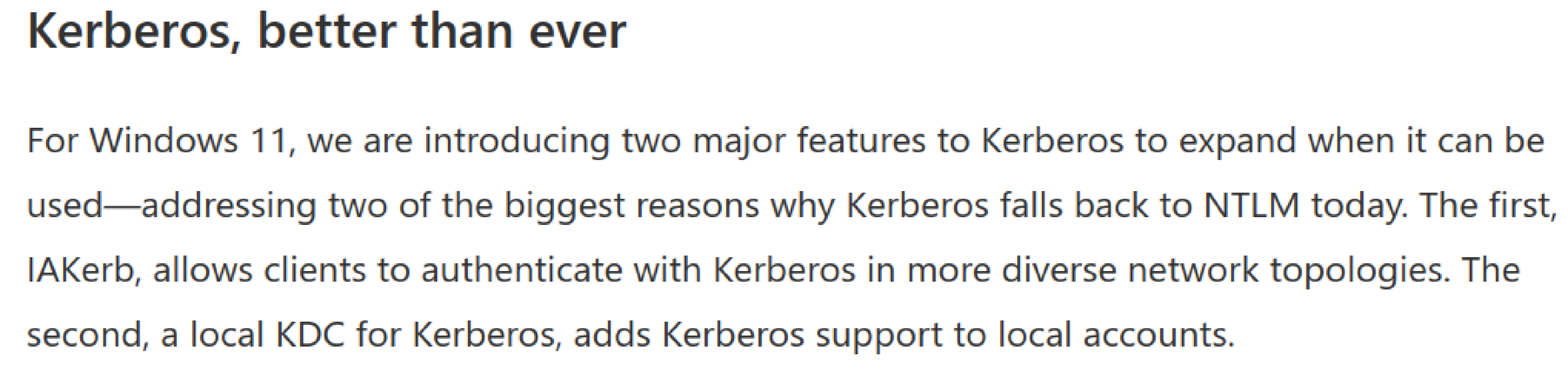
Why NTLM is so sticky?

NTLM

- doesn’t require local network connection to a Domain Controller (DC).
- is the only protocol supported when using local accounts.
- works when you don’t know who the target server is.

Full Circle In (Almost) 30 Years

- Microsoft (and everyone), since 2010 preaches:
  - Stop using NTLM!
- Microsoft, in 2023:



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.

3

The future (IAKerb)

Replacing NTLM Is Hard

- IAKerb and Local KDC are still not implemented on Windows 11 24H2 and Windows Server 2025

Initial and pass-through authentication using Kerberos (IAKerb)

- In 1997 MIT Kerberos folks designed a KRB5 proxy mechanism called IAKerb
  - Normally Kerberos requires three parties: a KDC, a client, and a service
  - What if the client can only have access to the service and not the KDC? (missing direct line of sight) => Solution: IAKerb

IAKerb in Kerberos implementations

- MIT Kerberos
  - IAKerb draft spec support was updated to be on par with Microsoft
    - not yet interop tested
    - Will be available with version 1.22

IAKerb in Kerberos implementations

- Heimdal
  - no IAKerb support yet (despite Apple's integration)

Apple and IAKerb (2007-2019)

- "Back to my Mac": Hey, you can have access to your own computer at home while travelling!

Apple and IAKerb (2007-2019)

- MacOS uses IAKerb and an external service to allow authentication back to your own machine
- The service was retired in 2019 but the bits are still present in the current MacOS version

4

Linux Local KDC

Local KDC

- A Kerberos KDC
  - Can be simulated as KDC bound to localhost (127/8) -- see our talk at FOSDEM 2024
  - Better to run on demand over UNIX domain sockets

Local KDC

- Access
  - is direct for a service (SSSD, Samba, SSH server, etc.)
  - or via IAKerb to clients (through service (proxy), if exposed)

Application service adjustment

- If GSSAPI is already supported
  - transparent locally (if app is built against MIT Kerberos)
  - just a special mechanism OID for IAKerb needs to be added

Application service adjustments

- Services which implement raw Kerberos calls
  - need to be compiled against MIT Kerberos
  - Can use the local KDC locally
  - IAKerb will not work unless converted to GSSAPI

Local KDC configuration

- The projects which provides the needed bits:
    - gitlab.com/cryptomilk/localkdc
    - Configures separate KDC in `/var/kerberos/localkdc`
    - Wraps `kadmin.local` for management as `localkdc-kadmin`
    - Relies on certmonger for PKINIT certs management
    - Fedora COPR: `dnf copr enable asn/localkdc` for quick demo/testing

5

Future work

MIT Kerberos

- IAKerb support (merged upstream)
- Stackable Kerberos KDC driver support (merged upstream)
- Kerberos principal aliases in local databases (in progress)

Samba ecosystem

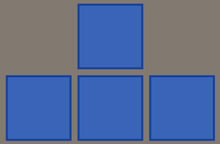
- IAKerb support in Samba (in progress)
- IAKerb support in cifs.ko/smb3.ko (in progress)

User database API integration

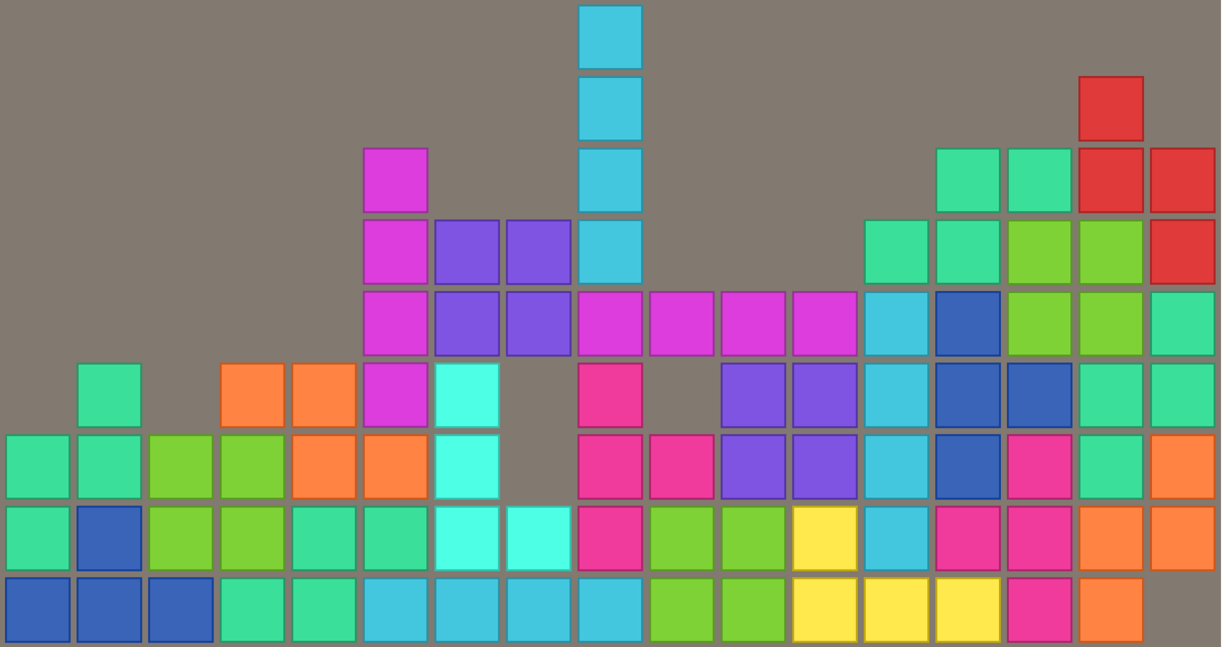
- Varlink support in KDB driver in MIT Kerberos (libarzirokim)
- Blend data from Kerberos KDB and systemd user database API
- Expose Samba-specific data in systemd user database
- Stackable PAC generator for local KDC

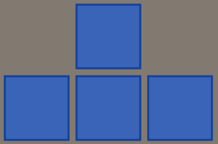
Passwordless methods support

- `ipa-otpd` and SSSD helpers for MIT Kerberos
- Varlink support in `ipa-otpd`



GAME OVER





Questions?

Mastodon (Alexander): @abbra:mastodon.social

Mastodon (Andreas): @cryptomilk:mastodon.social

Blog (Alexander): vda.li/en/ Blog (Andreas): blog.cryptomilk.org