Unleashing authentication for the Linux CIFS client with gssproxy

Daniel Kobras (PITC)
Michael Weiser (s+c)

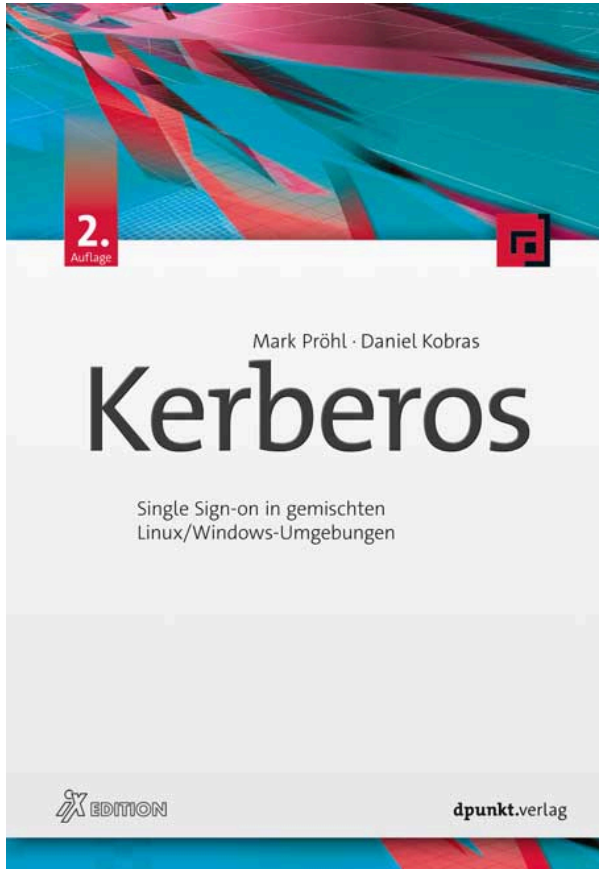
2025-04-07 • Göttingen • sambaXP

About us

Daniel Kobras

Principal Architect (Puzzle ITC)
Kubernetes/Container, Digital Identity, Kerberos

kobras@puzzle-itc.de



Michael Weiser

Senior Solution Architect (science+computing ag)
Kubernetes/Container, Cloud, Security, HPC

michael.weiser@eviden.com

science + computing

- IT service company in Germany (Tübingen, München, Berlin and Düsseldorf)
- Focus areas:
  - HPC
  - Big Data
  - Security
- https://www.science-computing.de/

PUZZLE ITC

changing IT for the better

- IT service company in Switzerland (Bern, Zürich, Basel) and Germany (Tübingen)
- Focus areas:
  - Open Source Technologies
  - Application Development
  - Container Plattforms, CI/CD
  - Linux System Engineering
  - Mobility
- https://www.puzzle.ch/

Agenda

Linux SMB with gssproxy

Overview
Use cases and pitfalls
Delegation and impersonation
Implementation
Demo

Introduction

SMB as general purpose Linux filesystem

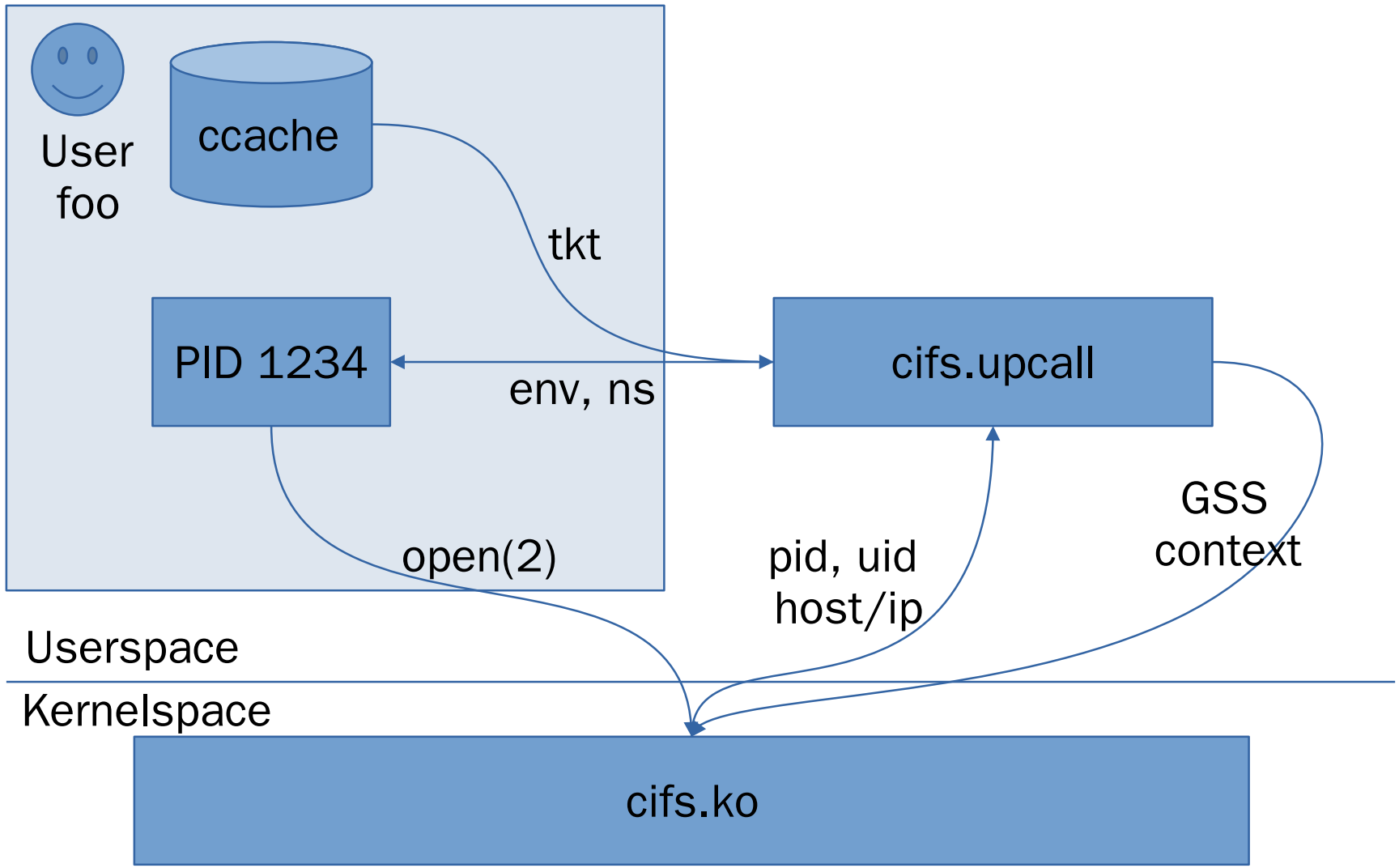
- Volker's mission: implement specs and functional requirements to establish SMB as an alternative to NFS
- Practical consequences:
  - SMB authorization requires authentication (unlike NFS/`AUTH_SYS`)
  - general purpose fs must support multi-user access
- cifscreds limited to password-based authentication
- remaining option: Kerberos + multiuser (aka multisession) mounts

SMB multiuser authentication with Kerberos

- Linux tasks and FS syscalls have no notion of Kerberos credentials
- Kerberos tickets and libraries live in userspace, FS client implemented in kernel

> *I went for a ticket, but all I got was this lousy ~~t shirt~~UID*
> *(`cifs.ko` never said that. But should do.)*

Obtaining credentials from upcall (simplified)



Upcall magic

- Clones namespaces from pid
- Clones `KRB5CCNAME` from pid's environment
- Otherwise uses `creduid` (fsuid) to derive default ccache (per `krb5.conf`)
- Derives service principal name from server hostname or IP address

> *`cifs.upcall` tries hard to get it right, but still fails in some cases.*

Selected subtle pitfalls

- How to obtain tickets during early login (eg. for `/home` on SMB)?
  - beware of parallel sessions w/per-session krb5 credentials
  - `systemd --user` processes not tied to session
  - ccache present, but `KRB5CCNAME` not yet initialized
- Generic, reliable operation requires
  - use of default ccache
  - per-user rather than per-session default
  - also see discussion in https://github.com/systemd/systemd/issues/7261

Obvious pitfalls

- Passwordless logins (eg. SSH keys)
- long-running sessions (exceeding ticket lifetime)
- cronjobs
- HPC/batch jobs
- systemd linger sessions

> *How to authenticate a user who isn't there?*

Interim summary

- Using SMB as a general-purpose FS on Linux in practice requires multiuser mounts with Kerberos authentication
- Strong user authentication introduces restrictions and challenges, especially for non-interactive access, and during early login

> *Is strong authentication of **individual users** a hard requirement at all?*
> *What if SMB only required strong authentication of **clients**, and trusted their user authentication?*

Impersonation and Delegation

Impersonation/Delegation with Kerberos

- (SMB) client is able to act as arbitrary user's identity
- Does not require active cooperation from affected users
- Ideally limited to selected (SMB) servers (constrained delegation)
- Available with (but not limited to) Active Directory (including Samba AD)
- Similar concept to NFS with `AUTH_SYS`, but stronger client authentication, auditability, and protection

> *More general alternative to SMB3's*
> *`SMB2_REMOTED_IDENTITY_TREE_CONNECT`*

Impersonation v1: Protocol Transition

- Configuration with Active Directory:
  - Set flag `TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION` (`0x01000000`) in attribute `userAccountControl` of computer object of SMB client
  - Add `cifs` service principalnames of all SMB servers to attribute `msDS-AllowedToDelegateTo` of SMB clients
- Properties:
  - Also works with older Kerberos libraries
  - Restricted to single domain/realm
  - Requires changes to computer object for each SMB client

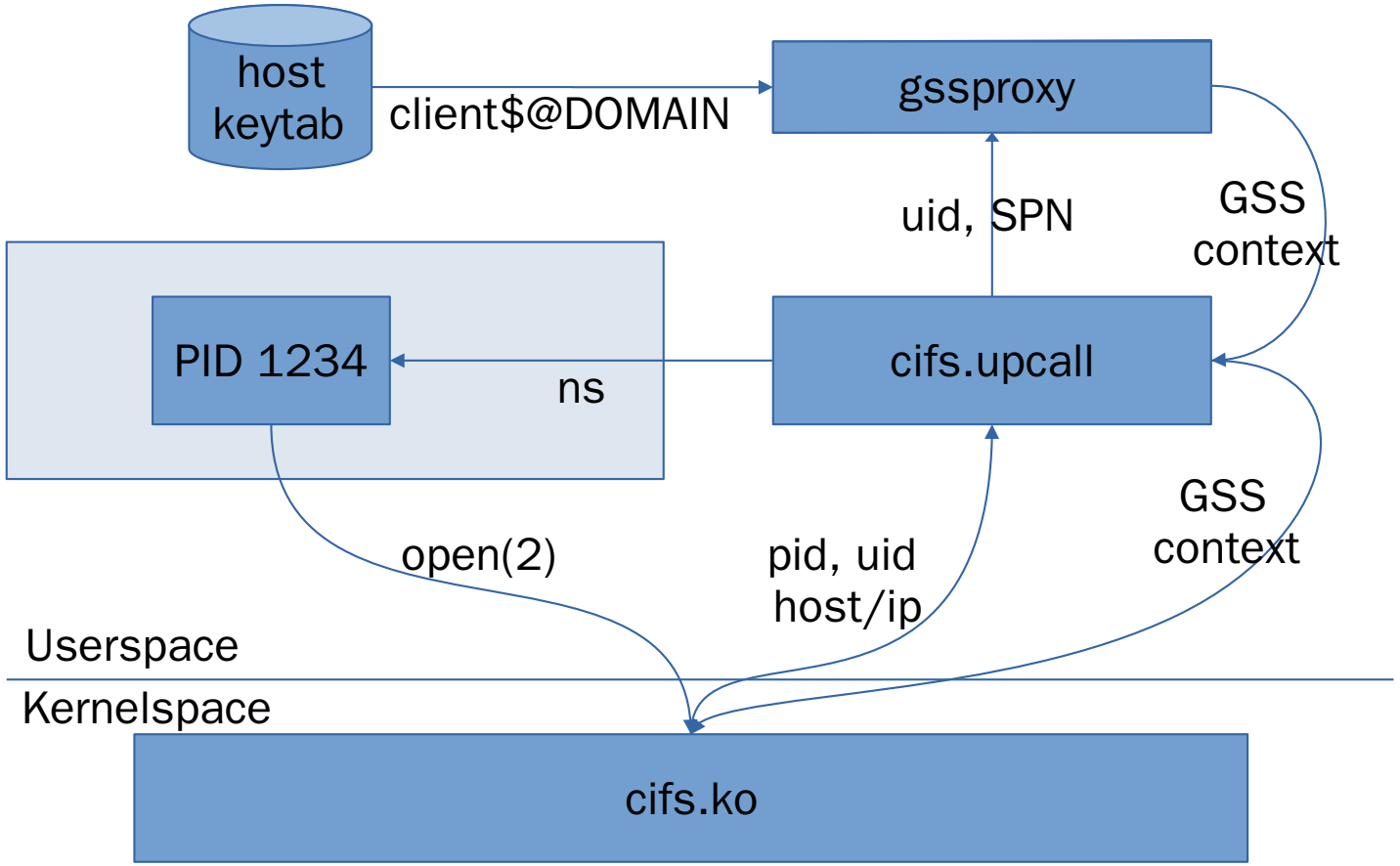
Impersonation v2: Resource-based Constrained Delegation

- No `forwardable` flag in S4U2Proxy requests required
- SID `S-1-18-2` (`SERVICE_ASSERTED_IDENTIY`) in `LOGON_INFO->SIDS` array of PAC identifies S4U2Self-issued tickets
- Configuration with Active Directory:
  - Add NT Security Descriptor for SMB clients to attribute `msDS-AllowedToActOnBehalfOfOtherIdentity` of SMB server's AD object
- Samba AD support since 4.17 (MIT)/4.19 (Heimdal)
- Properties:
  - Requires SMB clients with MIT Kerberos 1.19 or later (EL9, Ubuntu 22.04)
  - Works across multiple (trusted) domains
  - Only requires changes to SMB server's AD object
  - Requires tooling to construct (binary) security descriptors

Impersonation with Linux

- Option 1: Applications implement delegation support themselves
- Option 2: Applications implement Kerberos support using GSSAPI library
- Any GSSAPI-based application gets delegation support for free from `gssproxy` daemon:
  - `libgssapi` interposer to transparently hook proxy into library calls
  - optionally enable impersonation via `S4U2Self`/`S4U2Proxy`

Obtaining credentials with gssproxy (simplified)



Implementation

Configure gssproxy for SMB

- Install packages `gssproxy` and `cifs-utils` >= 7.0 on SMB clients
- static mount with machine identity in keytab:

```
mount -o vers=3.11,sec=krb5i,seal,multiuser,user='FS1$' //fs1.example.com/share /mnt
```

- or automount with user identity:

```
share -fstype=cifs,vers=3.11,sec=krb5i,seal,multiuser,cruid=${UID} ://fs1.example.com/share
```

- set env var `GSS_USE_PROXY` for upcall in `/etc/request-key.d/cifs.spnego.conf`:

```
create  cifs.spnego    * * /bin/env GSS_USE_PROXY=yes /usr/sbin/cifs.upcall %k
```

Impersonation with gssproxy

- Allow impersonation in `gssproxy` config for network-fs-clients: (`99-network-fs-clients.conf`)

```
[service/network-fs-clients]
(...)
allow_any_uid = yes
trusted = yes
impersonate = true  # <-- add this line
euid = 0
```

- Only with Protocol Transition (v1):
  request forwardable tickets by default (`/etc/krb5.conf`)

```
[libdefaults]
(...)
forwardable = true
```

Demo

Configuration Tips&Tricks

- Log impersonated principals
  - Increase `gssproxy` log level (`/etc/gssproxy/gssproxy.conf`)

```
[gssproxy]
debug_level = 1
```

- Impersonate as fallback only, if traditional Kerberos authentication fails
  - Set env var `GSSPROXY_BEHAVIOR=LOCAL_FIRST`
  - Not useful with current unmodified `cifs.upcall` because GSS code skipped if ccache was found

- Separate SMB from NFS config for gssproxy
  - Section `service/network-fs-clients` applies to NFS and SMB clients by default
  - `program` directive allows distinction, eg.
  - `/etc/gssproxy/99-nfs-client.conf`:

```
[service/nfs-client]
impersonate = false
program = /usr/sbin/rpc.gssd
```

`/etc/gssproxy/99-cifs-client.conf`:

```
[service/smb-client]
impersonate = true
program = /usr/sbin/cifs.upcall
```

  - Alternatively use separate UNIX domain sockets (directive `socket` and env var `GSSPROXY_SOCKET`)

Configuration Tips&Tricks

- AD accounts can be exempted from delegation (via group `Protected Users` or UAC flag `0x100000 NOT_DELEGATED`)
  - interactive access still possible (with valid Kerberos credentials)
  - authenticate from keytab (`/var/lib/gssproxy/clients/31337.keytab`) with override entry like `/etc/gssproxy/00-protecteduser.conf`:

```
[service/protecteduser]
mechs = krb5
cred_store = ccache:FILE:/var/lib/gssproxy/clients/krb5cc_%U
cred_store = client_keytab:/var/lib/gssproxy/clients/%U.keytab
cred_usage = initiate
# uid of protecteduser
euid = 31337
```

  - limits impersonation for these accounts to distinct clients with dedicated user keytab

Configuration Tips&Tricks

- Local accounts without associated user in AD:
  - configure explicit mapping to AD principal for SMB access
  - override entry like `/etc/gssproxy/00-localuser.conf`:

```
[service/localuser]
mechs = krb5
cred_store = keytab:/etc/krb5.keytab
cred_store = ccache:FILE:/var/lib/gssproxy/clients/krb5cc_%U
cred_usage = initiate
krb5_principal = aduser@EXAMPLE.COM
trusted = yes
impersonate = true
# uid of localuser
euid = 101
```

Summary

Keys Takeaways

- SMB as a general-purpose Linux filesystem has its pitfalls beyond actual filesystem features
- `gssproxy` and delegation allow to solve some of these problems, especially for non-interactive accesses
- GSSAPI support in `cifs.upcall` is essential to make use of `gssproxy` and its features

> *It's not a must to use gssproxy with SMB mounts.*
> *But it's a must to know about gssproxy, and how to use it.*

Thank you!

Michael Weiser michael.weiser@eviden.com

Daniel Kobras kobras@puzzle-itc.de