Integrating Storage Systems into Active Directory

SDC EMEA 2019
Tel Aviv

Volker Lendecke

Samba Team / SerNet

2019-01-30

Overview

- Active Directory
- Authentication Mechanisms
- Windows- and Unix-IDs
- API introduction

Who am I?

- Co-Founder of SerNet in Göttingen, Germany
- First Samba patches in 1994
- Early Samba Team member
- Samba infrastructure (tdb, tevent, etc)
- File server
- Clustered Samba
- Winbind
- AD controller is my colleague Stefan Metzmacher's domain
  - Stefan implemented AD multi-master replication in Samba

Active Directory

- Microsoft's central user database
  - Successor to NT4-based Security Account Manager (SAM)
  - It's what eDirectory is for the Bindery (Novell anyone?)
- Kerberos KDC with an LDAP database backend
- Multi-Master replicated LDAP database
- Highly specific LDAP schema with custom extensions
  - A lot of internal magic and validity checks
- Authentication server for Challenge-Response based schemes
- DNS database for server lookup
- Often very complex setup of many domains (Realms)
  - Cross-Realm authentication is common, not an exception

What is Samba?

- Started in the 1990s as a DEC Pathworks file server
- Originally based on Solaris
- Implementation of many Microsoft Protocols
  - Server Message Block (SMB) for file services
  - SMB and DCE-RPC for print services
  - RPC for user database services
  - Kerberos, DNS, LDAP, etc
- NT4 compatible Domain Controller
- Active Directory Domain Controller
- Active Directory Domain Member
  - Make AD users and groups available to Linux/Unix

Authentication and Authorization

- Did the user type in her/his password correctly?
- What is the user allowed to do?
  - What groups is the user member of?
  - What is the user's access token?
  - Access token in Windows Style or Unix Style?

Authentication mechanisms

- telnet/ftp: Not spending much time here
  - Salted and hashed passwords on the server's disk
- ssh: plain text passwords protected by public key crypto
  - Also public key authentication
- Challenge/Response
  - Server offers a Nonce, client encrypts nonce with user's password
  - Server does the same and compares the result
  - Plain text password on server's disk
- Kerberos: Complicated version of challenge/response
  - Plain text password on KDC disk

NTLM vs Kerberos

- NTLM
  - MS' Challenge Response Authentication Protocol flavor (a.k.a. CRAP)
  - Not as CRAP as it used to be, modern versions are resonably secure
  - For every authentication the DC must be asked
- Kerberos is the "standard" authentication protocol
  - Based on signed tickets with lifetimes
  - Reduced load on the DC due to ticket caching
  - Can be very picky, often fails
  - Server must be contacted by it's name, IP addresses don't work
- NTLM as a fallback must always be available

Roles in Authentication

- User
  - The one who knows a password, presents a certificate or similar
- Authenticating workstation or server
  - Machine a user requests access to
- Domain Controller / Key Distribution Center
  - Central user database, point of trust
  - Gatekeeper for all access control decisions
- Workstation/Server has to trust the DC
  - Trust based on a shared secret / workstation password
  - DC proves that it knows the workstation password
  - In Kerberos-speak that’s a machine principal and keytab

Samba's winbind

- Daemon responsible for all DC traffic
- Domain Controller lookup (DNS SRV records, CLDAP, NetBIOS)
- Establish encrypted and verified DC connection
- All nasty Microsoft RPC is done by winbind
- Machine password changed regularly
  - Maintenance of the trust account
- Very (too?) simple socket interface on /tmp/.winbindd/pipe
- Samba's PAM and NSS modules redirect to winbind
- Tries to do exactly what Windows clients do
  - That's all we can rely on
  - Not fully there yet though

Authorization

- Authentication is done via Kerberos or NTLM via winbind
- Authorization: What is the user allowed to do?
    - Utilize permissions from ACLs
    - ACLs are defined for User- and Group-IDs
- What is a user's UID and what groups is she/he member of?
- Domain Controllers are the ones to know group memberships
    - User token describes the user precisely
- AD only provides the Token upon successful authentication
    - Kerberos tickets and NTLM CRAP reply contain all user info

User token calculation

- Access control needs User ID and a list of Group IDs
- Active Directory has a very complex group model
  - Group Types: Domain, Universal, Domain Local, Local groups
- Group memberships can be nested
- Domain Controllers calculate membership at login time
  - Kerberos initial user login NTLM authentication
  - Winbind can't calculate group memberships for users not logged in
  - NFS –manage-gids not reliable
- Future development: Service4U2Self via Kerberos

ID mapping

- Windows user IDs are 128 bit
  - Under Windows, multiple domains are seamlessly integrated via trusts
- Unix user, groups and ACLs defined in terms of 32-bit values
  - Merging organizations is a nightmare
- Every Windows user and group needs a stable unix equivalent
  - Multiple servers need to agree
- Winbind has multiple ways to do idmapping
  - Static configuration per domain: idmap_rid
  - Table-based: idmap_autorid or idmap_tdb
  - AD-maintained mappings: idmap_ad

winbind nss info

- Active Directory can maintain Unix user information
- Services for Unix (SFU) schema extension
- Every user can get a uidNumber
- User objects have two primary groups: Windows and Unix
- Before 4.6, Samba only looked at the Windows primary group
- Some customers don't want a gidNumber for "Domain Users"
  - idmap config DOMAIN : unix primary group = yes
  - idmap config DOMAIN : unix nss info = yes

libwbclient

- Let’s look at some header file

Questions?

vl@samba.org / vl@sernet.de
http://www.sambaxp.org/