Implementing a Unified Login for Windows and Unix clients

Peter Gietz, DAASI International GmbH
Peter.gietz@daasi.de
Agenda

- Pre-requisites for our approach
  - DFN-Research project of DAASI International
  - Requirements
- Technologies needed
- Unified Login with Active Directory
- Unified Login with OpenLDAP/SAMBA
  - Why do we want to do that?
  - Problems
  - Solution
  - Zope based webgui
- Migration from Active Directory to OpenLDAP
- Experiments with SAMBA 3.0
Pre-requisites of our work

- DAASI International Ltd.
  - Directory Applications for Advanced Security and Information management
  - A spin-off of directory related research projects at University of Tübingen
  - Performed the BMBF funded DFN project „Ausbau und Weiterbetrieb eines Directory-Kompetenzzentrums“ (DFN Directory Services)

- Part of the project was to implement a Unified Login Service for a University environment
Aims

- A Unified Login Service
  - For the heterogenous environment of German Universities
  - For up to 40,000 users
  - Integrated in existing infrastructure
  - Scalable solution without performance loss
- Should lead to:
  - Reduction of system administration work
  - Reduction of Helpdesk effort
    - „I forgot my password“
  - => Reduction of costs
- Less passwords to remember should lead to stronger passwords
Requirements

- Basic operating system functions for user and group lookup
- User authentication for
  - Console logins (Unix and Windows)
  - Secure remote shells (SSH)
  - Email submission (SMTP) and retrieval (IMAP)
  - Email routing
  - Webpage access
- Integration with a white-pages service
- Passwords must not be send in clear text
- Enforcement of Password policy
- Single Sign On
Statistics

- Daily amount of emails and logins at a university computing centre
  - Up to 70,000 email to route per day (a historic peak was 220,000 emails on one day)
  - Up to 50,000 pop3 logins per day
  - Up to 25,000 IMAP logins per day
  - This amounts to 150,000 search requests and 80,000 authentication operations per day only for email services
Kerberos
- Network authentication protocol with strong authentication for client/server environments
- Each participant shares a secret key with a central Key Distribution Center (KDC)
- KDC consists of Authenticate Service and Ticket Granting Service

GSSAPI (Generic Security Service Application Program Interface)
- Security framework that abstracts from underlying protocols
- Includes a Kerberos mechanism
Useful Technologies 2

- X.509
  - Certificate based strong authentication via asymmetric encryption
  - Certificate issued by a third trusted party (CA)

- Security Layers
  - Integrity and privacy protection via encryption
  - Secure Socket Layer (SSL) / Transport Layer Security (TLS)
    - X.509 Certificate based
  - Kerberos and SASL also can establish Security Layers
  - IPSec: X.509 certificate based security at the network layer
SASL (Simple Authentication and Security Layer)
- Method for adding authentication support to connection-based protocols
- Supported by LDAP Servers
- Specified mechanisms:
  - PLAIN (plain text password, we don’t want that!)
  - DIGEST-MD5 (challenge Response no clear text PW)
  - GSSAPI (and thus Kerberos)
  - EXTERNAL (e.g. X.509 certificate used in the underlying SSL / TLS)
Useful Technologies 4

- **Name Service Switch (NSS)**
  - Layer in Unix C libraries that provides different means for listing or searching users, groups, IP services, networks, etc.:
    - Flat files (etc/passwd, etc.) = hard to administrate
    - NIS (Network Information Service) = security holes
    - LDAP = 😊

- **Pluggable Authentication Modules (PAM)**
  - Framework for login services
  - Manages authentication, accounts, sessions and passwords
  - Modules exist for LDAP, Kerberos, etc.
Unix authentication

- Application
- C library
- NSS library
- flat files
- passwd
- PAM library
- LDAP
- NIS
- SMB

Directory Applications for Advanced Security and Information Management

DAASI International
Very useful technology 😊

- LDAP (Lightweight Directory Access Protocol)
- It is a database or information model (X.500)
  - Hierarchical structure
  - Object oriented
  - Extensible for any kind of data
- It is a network protocol
  - Internet standard
  - Client/server
  - Flexibly extensible
  - Allows for distribution of data in the net (just like WWW!)
  - Allows for replication of the data in the net
Unified Login with Active Directory (AD)

- First project result was based on AD
  - Usefull in a primarily Windows based landscape
  - Integrated Kerberos Key Distribution Center (KDC) easily provides SSO functionality
  - AD did not fully support NIS schema,
    - Open LDAP server was additionally used for NIS data
      - AD was only used for authentication
  - PAM_LDAP as well as PAM_krb5 could be used, easily switchable
  - SSO system supports Unix and Windows login, SMTP auth, IMAP auth, SSH, CVS, FTP
Why search for something else?

- We needed a more flexible solution
  - something in which you can integrate your own code => Open Source
- No licensing problems
- Better Unix support
- Only one directory for all applications
  - Not only integrate NIS but any directory services
  - Easier administration
    - One central administration point
    - Different admins have different access rights (on subtree and on attribute level)
    - Good old log files instead of strange error messages
- Easier replication mechanism
Example integration into other services

- IMAP server
- LDAP-master
- LDAP login-server
- LDAP email directory
- LDAP telefon directory
- LDAP university calender
- administration interface 1
- administration interface 2

Intranet

DMZ

Data management replication

web gateway

web gateway

web gateway
OpenLDAP/Samba recipe

- Take a linux box with minimal linux installation
- Add the following (newer versions will also do):
  - binutils-2.11.90.0.29-15.i386.rpm
  - gcc-2.95.3 136.i386.rpm
  - glibc-devel-2.2.4-40.i386.rpm
  - make-3.79.1-180.i386.rpm
  - nss_ldap-167-54.i386.rpm
  - openldap2-2.0.12-33.i386.rpm
  - openldap2-client-2.0.12-28.i386.rpm
  - openldap2-devel-2.0.12-28.i386.rpm
  - openssl-devel-0.9.6b-62.i386.rpm
  - pam-devel-0.75-78.i386.rpm pam_
  - ldap-122-77.i386.rpm
- And don’t forget Samba, we took 2.2.8a
- Useful are the IDEALX smbdap-tools-0.7.tgz
The big picture
Client platforms that work

- Unix:
  - Linux
  - FreeBSD
  - OpenBSD
  - NetBSD
  - Solaris
  - HP-UX
  - AIX

- Windows:
  - 2000
  - XP
Production service

- We currently use central authentication for:
  - Linux client login
  - BSD client login
  - Win2k client login
  - Cyrus-imapd
  - Sendmail smtp auth
  - sshd
  - cyrus-sasl
  - tutos (open source project planner / CRM)

- We do cashing via Name Service Casheing Daemon (nscd)
Problems

- Memory allocation reentrance bug in SASL made the following authentication chain crash: cyrus-imapd -> cyrus-sasl -> pam -> pam_ldap
- Either redesign the SASL library (⊗) or use the work around patch of Rein Tollevik
Zope based user/admin interface

- Easy to use interface for users and admins
- Using Zope
  - Very portable
  - Nice CMS functions
  - Has an LDAP API ("LDAPUserFolder")
- Interface uses SSL/TLS
- Manages any kind of data
Change the basic properties of your LDAPUserFolder on this form.

<table>
<thead>
<tr>
<th>Title</th>
<th>Zentrale Authentifikation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name Attribute</td>
<td>uid (uid)</td>
</tr>
<tr>
<td>RDN Attribute</td>
<td>uid (uid)</td>
</tr>
<tr>
<td>Users Base DN</td>
<td>ou=Users, o=smb, dc=daasi, dc=de</td>
</tr>
<tr>
<td>Group storage</td>
<td>Groups stored on LDAP server</td>
</tr>
<tr>
<td>Groups Base DN</td>
<td>ou=Groups, o=smb, dc=daasi, dc=de</td>
</tr>
<tr>
<td>Manager DN</td>
<td>cn=root, o=smb, dc=daasi, dc=de</td>
</tr>
<tr>
<td>Manager DN Usage</td>
<td>For login data lookup only</td>
</tr>
<tr>
<td>User object classes</td>
<td>top, inetOrgPerson, posixAccount, sambaAccount</td>
</tr>
<tr>
<td>User password encryption</td>
<td>SHA</td>
</tr>
<tr>
<td>Default User Roles</td>
<td>Anonymous</td>
</tr>
</tbody>
</table>
Adding or removing attributes on this page does not affect your LDAP schema in any way, it will only affect what the LDAPUserFolder knows about your schema.

<table>
<thead>
<tr>
<th>LDAP Attribute Name</th>
<th>Friendly Name</th>
<th>Mapped to Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>gecos</td>
<td>(Posix) GECOS</td>
<td></td>
</tr>
<tr>
<td>gidNumber</td>
<td>(Posix) GID Number</td>
<td></td>
</tr>
<tr>
<td>homeDirectory</td>
<td>(Posix) Home Directory</td>
<td></td>
</tr>
<tr>
<td>loginShell</td>
<td>(Posix) Preferred Login shell</td>
<td></td>
</tr>
<tr>
<td>uidNumber</td>
<td>(Posix) UID Number</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>(Win) Description of the user</td>
<td></td>
</tr>
<tr>
<td>profilePath</td>
<td>(Win) Path to profile</td>
<td></td>
</tr>
<tr>
<td>smbHome</td>
<td>(Win) Path to server homes</td>
<td></td>
</tr>
<tr>
<td>scriptPath</td>
<td>(Win) Path to startup script</td>
<td></td>
</tr>
<tr>
<td>rid</td>
<td>(Win) Relative ID</td>
<td></td>
</tr>
<tr>
<td>displayName</td>
<td>(Win) displayed name of user</td>
<td></td>
</tr>
<tr>
<td>cn</td>
<td>Canonical Name</td>
<td></td>
</tr>
<tr>
<td>givenName</td>
<td>Given name</td>
<td></td>
</tr>
<tr>
<td>sn</td>
<td>Last Name</td>
<td></td>
</tr>
<tr>
<td>telephoneNumber</td>
<td>Telefonnummer</td>
<td></td>
</tr>
<tr>
<td>uid</td>
<td>uid</td>
<td></td>
</tr>
</tbody>
</table>
**Benutzerdetails**

- **DN:** `uid=seb, ou=Users, o=smb, dc=daasi, dc=de`
- **cn:** Sebastian Stark
- **givenName:** Sebastian
- **gecos:** Sebastian Stark
- **loginShell:** `/bin/bash`
- **dn:** `uid=seb, ou=Users, o=smb, dc=daasi, dc=de`
- **telephoneNumber:** 4321
- **uid:** seb
- **displayName:** Sebastian Stark
- **sn:** Stark

**Passwort**

| Neues Passwort: | 
| Bestätigung (Neues Passwort bitte nochmal eingeben): | 
| Hash: | 

Submit
Migration from AD to OpenLDAP

- IDEALX tools help to migrate passwords
- We wrote a script that migrates all infos stored in AD to the OpenLDAP server
- You can in theory also migrate the profiles since samba supports the roaming profile feature (we are still working on that)
Results

- Stable service via replicated LDAP server
- No performance problems via caching
- Both directory implementations (AD and OpenLDAP) are fast enough for the requirements of a university
Pros and cons

Advantages:
- User remembers only one password
- Admin‘s and helpdesk‘s life is far easier
- Unification of authentication processes
- Central point for password evaluation
- Before implementation you need a concept

Caveats:
- single point of failure (if without replication)
- You need to enforce password policy (not yet implemented in OpenLDAP)
- Admin access to clients should use local passwords
Our view on Samba 3.0

- The "ldap passwd sync" feature main reason to switch to Samba 3.0.
  - Users can change their password using the standard windows password change dialog.
  - Samba cares for the necessary steps to update both, the passwords used by windows (LDAP attributes: ntPassword and lmPassword) and the userPassword attribute that is used by Unix clients.
  - Samba can delete a complete dn if the user is to be deleted from the Samba account database (= ldapsam) or only remove the attributes concerning windows.
Samba 3.0 (contd.)

- The "ldap trust ids" feature
  - assumes that user ids returned from the LDAP database are always correct
  - So no need to lookup the corresponding Unix user.
  - This is very useful for our setup since we use nss_ldap and thus have valid UIDs in our database anyway.

- The upgrade process was clean and easy.
  - Having the account data in an LDAP directory does really help this process.

- Now the Code must prove its stability in our production environment.

- Not yet experimented with:
  - PDC replication stuff to set up a multimaster environment with Samba.
  - Samba Active Directory emulation.
  - group mapping of Samba 3.0 (still incomplete ?)
Where to go from here?

- Use Samba 3.0 in production service
- We are about to include SSO functionality via Kerberos
- Password policy in OpenLDAP!
- What about a complete domain controller simulation via Samba?
  - AD replication!
References

- RFC 1510, „The Kerberos Network Authentication Service (V5)“
- RFC 1964, „The Kerberos Version 5 GSS-API Mechanism“
- RFC 2222, „Simple Authentication and Security Layer (SASL)“
- RFC 2246, „The TLS Protocol Version 1.0“
- RFC 2307, „An Approach for Using LDAP as a Network Information Service“
- RFC 2743, „Generic Security Service Application Program Interface Version 2, Update 1“
- RFC 2829, „Authentication Methods for LDAP“
- RFC 2831, „Using Digest Authentication as a SASL Mechanism“
More references

- Samba: www.samba.org
  - IDEALX tools: www.idealx.org/prj/samba/index.en.html
- LDAP:
  - New drafts: www.ietf.org/html.charters/ldapbis-charter.html
  - OpenLDAP: www.openldap.org
  - NSS_LDAP: www.padtel.com/OSS/nss_ldap.html
  - PAM_LDAP: www.padtel.com/OSS/pam_ldap.html
  - Reentry patch from Rein Tollevik: www.openldap.org/lists/openldap-software/200108/msg00594.html
- X.509:
  - www.ietf.org/html.charters/pkix-charter.html
- Cyrus project (SASL, IMAP): asg.web.cmu.edu/cyrus/
- Zope: www.zope.org
- Tutos: www.tutos.org
Yet one more reference

- Diploma thesis on the subject, which was made inside the project:

Norbert Klasen: „Directory Services for Linux in comparison with Novell NDS and Microsoft Active Directory“, www.daasi.de/staff/norbert/thesis/
THANKS FOR YOUR ATTENTION

- DAASI International
  - http://www.daasi.de
  - Info@daasi.de

- DFN Directory Services
  - http://www.directory.dfn.de
  - Info@directory.dfn.de