

Open and Closed Source ID Management Integration

Hege Trosvik, University of Oslo, USIT

Abstract

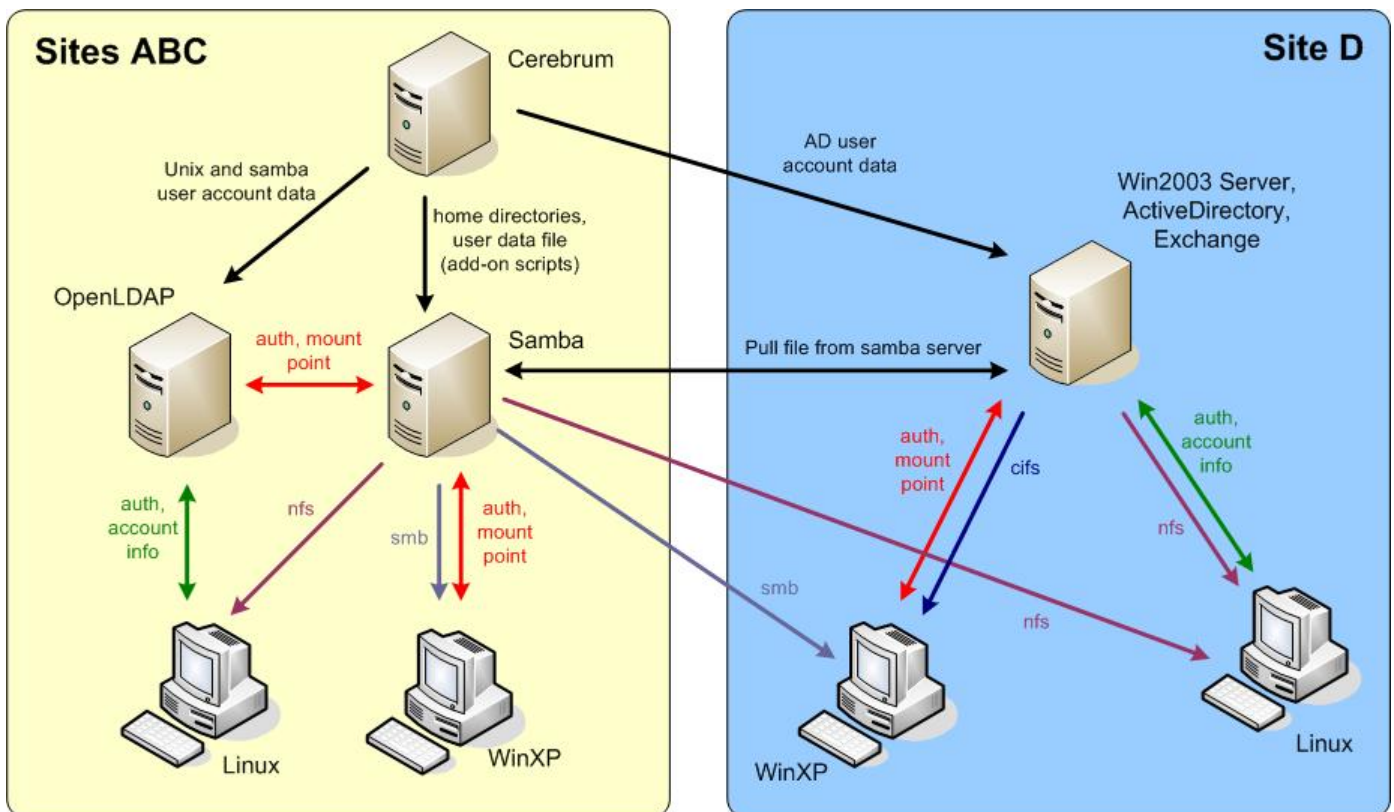
As Linux and other open source operating systems and applications are gaining in popularity, the biggest challenge presented to enterprises wishing to run a mixed environment is how to handle user authentication, authorization and various kinds of user related data, also known as *Identity Management* or *ID Management*. This includes the administration of user home directory, email and calendaring information. Authenticating users on both Linux and closed source platforms against common servers and directories is best case not trivial and worst case impossible.

In the iLabs Open Source Software (OSS) initiative, we test and demonstrate this integration and a number of the different options available. Our building blocks include Samba, OpenLDAP, Active Directory (AD), OpenXchange and Exchange. We have both open and closed source clients authenticating to, mounting home directories from and sending and receiving email to and from both open and closed source servers. This paper gives a brief description of our solution with examples of the various client authentication methods and file server options we have implemented as well as some of the challenges we presented us with.

Network Topology

Our lab network simulates a company with three sites (A, B, C) running open source operating systems and applications on all servers and the majority of desktops, to show how a complete infrastructure can be implemented with open source software. A fourth, newly acquired site (D), had an infrastructure based almost entirely on closed source software at the time of the acquisition. This allows us to show the integration of open source software in ABC with the closed source infrastructure in D. **Figure 1** shows a schematic drawing of our ID management infrastructure, the various pieces will be explained in more detail in the following sections.

Figure 1 ID management infrastructure



Cerebrum

Cerebrum is the core application in our ID management infrastructure. In our demo, Cerebrum presents the user with a web interface where he can create his own user account for the purpose of our demonstration. The user enters his name and his imaginary role in our virtual company through the web interface. Cerebrum generates the required data for the user account and performs the following tasks:

- Sends information related to the organization (defined ou's), people associated with the organization, users (name, username, email address, dn, password (NT-hash for samba, MD5, clear text), UID, GID, homedirectory, various samba information, dataName, UID, GID, password, OpenXchange attributes) and groups (name, description, GID, members, samba info) to OpenLDAP
- Sends similar information to OpenXchange to keep the postgres database on OpenXchange in sync with OpenLDAP
- Sends user account related data to an ADSI service on the Windows server which creates the user account in AD
- Triggers a script that creates the home directory for the user on the samba server and copies a file with the information necessary in the required format to a specific share on the Samba server. A script on the Windows server then reads this file and creates a home directory and an email account in Exchange for the user. Since the frequency of creating new users is fairly low, the service checks for updates every 5 seconds to avoid noticeable delay in the propagation of the user data.

Cerebrum runs on the server shown at the top in **figure 1**. Sites ABC is the open source part of the company, the Cerebrum server is located here together with the OpenLDAP and Samba servers. Site D is the closed source part of the company, the Windows2003 server with AD and Exchange is located on site D. The black arrows show how Cerebrum pushes user data out to the various servers as described above.

OpenLDAP

OpenLDAP is the LDAP directory where we store our user information for our open source world. OpenLDAP is part of the FC4 distribution. We had to collect the LDAP schemes required by Cerebrum from various places, place them under `/etc/openldap/schemas/` and include them in the `slapd.conf` file. We also had to generate the `slapd` password for the `slapd.conf` file with the `slappasswd` command. Other than that configuration was straight forward. **Figure 2** shows the non-default parts of the `slapd.conf` file.

As a simple way of testing that the OpenLDAP server was working, we used the ldap client utility `ldapmodify` to push a simple ldif file creating our oss domain, organizational unit (ou) people and unix user oss into openldap. This command is shown in **figure 3**, the ldif file is shown in **figure 4**. The oss user has a number of attributes that are necessary for a unix account, including UID, GID, home directory and password as well as the object classes `posixAccount` and `shadowAccount`. These are defined in `nis.schema` which is included in OpenLDAP by default.

```
[root@a-1 tmp]# more /etc/openldap/slapd.conf

include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/samba.schema
include /etc/openldap/schema/eduperson.schema
include /etc/openldap/schema/eduorg.schema
include /etc/openldap/schema/FEIDEldap.schema
include /etc/openldap/schema/openxchange.schema
include /etc/openldap/schema/openxchange-admin.schema
include /etc/openldap/schema/mail.ldap.uio.no.schema

database bdb
suffix "o=oss.ilabs.interop.net"
rootdn "cn=manager,o=oss.ilabs.interop.net"
rootpw {SSHA}VJzc+feRt353G+NZh6wozDc9oA8qRzS0
```

Figure 2 slapd.conf

```
[root@a-1 ~]# ldapmodify -a -x -D
"cn=manager,o=oss.ilabs.interop.net" -c -W < /tmp/oss.ldiff
Enter LDAP Password:
adding new entry "o=oss.ilabs.interop.net"
adding new entry "ou=People,o=oss.ilabs.interop.net"
adding new entry "uid=oss,ou=people,o=oss.ilabs.interop.net"
```

Figure 3 ldapmodify

is required to browse attributes, however to edit you need to enter the management user and password configured for the LDAP root.

The LDAP browser from <http://www-unix.mcs.anl.gov/~gawor/ldap/download.html> is quite handy for setting and reading LDAP attributes. The LDAP browser is written in java and just needs to be untar'ed into a directory, have `JAVA_HOME` set and is ready to run with `./lbe.sh`. To access a specific LDAP server, it needs the server address and the base dn (distinguish name). No password

Linux authentication against LDAP

On the Linux client PAM (Pluggable Authentication Modules) is the key component for authentication. There are three files that need to be modified to have the linux client get its authentication data from OpenLDAP. On FC4 `/usr/sbin/authconfig` sets most of this, but check that it configured what you wanted. The first file that needs to be modified is `/etc/ldap.conf`, it needs to have the IP address for the ldap server and the base dn.

The second file is `/etc/nsswitch.conf`, that needs to know it should search LDAP for passwords and groups.

The third file, `/etc/pam.d/system-auth`, also needs to be configured to use LDAP. This file is auto-generated by `authconfig`, so you shouldn't have to worry about this one as long as you initially use `authconfig` to enable LDAP for authentication. **Figure 5** shows the critical lines in these files.

The home directory is already mounted on the linux client via NFS, so no additional magic is required for this client as soon as Cerebrum has pushed the correct user data to OpenLDAP and the script has created the home directory for the user on the Samba/NFS server. When the user logs in, the PAM module requests the info necessary to authenticate the user from OpenLDAP and if the password entered matches, he is allowed to log on and can access his home directory. The manually created oss user can be used to test that this when it is desirable to test the various building blocks independently. For test purposes it may be a good idea to adjust the timeout values in `ldap.conf`.

Figure 4 oss.ldif

```
[root@a-1 tmp]# more oss.ldif
dn: o=oss.ilabs.interop.net
dc: oss
objectclass: top
objectclass: domain
o: oss.ilabs.interop.net

dn: ou=People,o=oss.ilabs.interop.net
objectclass: top
objectclass: organizationalUnit
ou: People

dn: uid=oss,ou=people,o=oss.ilabs.interop.net
objectclass: top
objectClass: account
objectClass: posixaccount
objectClass: shadowaccount
cn: oss
uid: oss
uidNumber: 2001
gidNumber: 2000
homeDirectory: /home/oss
userPassword: MyPassw0rd
loginShell: /bin/bash
```

```
[root@localhost ~]# more /etc/ldap.conf
host 45.230.10.101
base o=a.oss.ilabs.interop.net
ssl no
pam_password md5

[root@localhost ~]# grep ldap /etc/nsswitch.conf
passwd:      files ldap
shadow:      files ldap
group:       files ldap

[root@localhost ~]# grep ldap /etc/pam.d/system-auth
auth         sufficient /lib/security/$ISA/pam_ldap.so use_first_pass
password     sufficient /lib/security/$ISA/pam_ldap.so use_authtok
session      optional   /lib/security/$ISA/pam_ldap.so
```

Figure 5 ldap.conf, nsswitch.conf, system-auth

Samba and NFS

The samba server is our file server for both linux (NFS) and windows (SMB/CIFS) and also authenticates windows clients in the open source part of our network. NFS is part of most Linux distributions and well known to most system administrators. We chose to have our `/home` directory permanently mounted on our Linux clients to keep our setup simple. Samba is also part of the FC4 distribution and fairly easy to configure, although a little understanding of both open and closed source and how the samba server interacts with both worlds does indeed help.

Our samba server is configured as a NT4-style primary domain controller for `oss.ilabs.interop.net`, and does not interact with the Windows domain controller directly. The samba server uses OpenLDAP as a backend for all user data. One way to achieve this is to configure the samba server to be an LDAP client as described above and let it know where in the LDAP tree to find various pieces of information. **Figure 6** shows the samba config file, `/etc/samba/smb.conf`. The password for the LDAP admin user is stored in the `secrets.tdb` file which is created by running `smbpasswd -w <passwd>`. The samba server is configured with the special `homes` share to allow users to mount their home directories and a public share `documents` for sharing files between users.

We did not enable TLS (Transport Layer Security) since securing this implementation was not our highest priority, but rather left to the end should we have time. This was the case across most of our applications. Since our LANs, except from the management VLAN, were not routed outside of the show floor, security was not our main concern.

The OpenLDAP server needs to have `samba.schema` which comes with samba and the objectClass `sambaSamAccount`, and the attributes `sambaSID` and `sambaNTPassword` for all samba users and it also needs to have entries for the `sambaDomainName` and the `sambaSID` and an LDAP admin user for samba. The client PCs need to have a trust account to authenticate the PC itself to the domain controller. We used `smbldap-tools` to add client PCs to LDAP.

Windows authentication against Samba

Setting up windows PCs to authenticate against Samba with an LDAP backend is also fairly simple. The PC must be member of the Samba server's domain. To make the client a member of the domain, administrator rights for the domain is required.

Active Directory

The good thing about Active Directory (AD) is that it is mostly just point and click. The bad thing about AD is also that it is just point and click, it is not always trivial to figure out what is set where and to find debugging information. There is also frequently a lot of clicking to find what you look for. If you are used to dealing with Microsoft products, however, the AD interface is fairly intuitive, or at least has a familiar look and feel similar to many other Microsoft products.

We configured our Win2003 server with Active Directory (and Exchange) to be the domain controller for domain d.oss.ilabs.interop.net. Cerebrum pushes user data out to a service on the Windows2003 server and the service creates an AD account for the user. Additional data the server needs to create the user home directory and exchange account is copied in a file per user from the samba server as described above.

Windows authentication against Active Directory

The windows client in D is configured to be part of the domain b.oss.ilabs.interop.net and is authenticated against Active Directory when logging on to the domain. The client needs to be a member of and trusted by the domain to be able to authenticate users in the domain and allow them access to the PC if the AD server verifies username and password.

Administrative rights in the domain are required to make the client a member of the domain. Kerberos is used for user authentication in the domain and when authenticated, the user gets a kerberos token from the domain controller.

The client also stores username and password information locally (stored credentials) so that if a user attempts to access network services he is not authenticated for, the client will try to pass this information along automatically.

```
[root@samba samba]# more /etc/samba/smb.conf
[global]
  workgroup = OSS
  netbios name = SAMBA
  server string = OSS server
  passdb backend = ldapsam:ldap://45.230.10.101
  username map = /etc/samba/smbusers
  add machine script = /usr/sbin/smbldap-useradd -w '%u'
  logon script = scripts\logon.bat
  logon path = \\SAMBA\homes\profiles
  logon drive = M:
  logon home = \\SAMBA\homes
  domain logons = Yes
  os level = 35
  preferred master = Yes
  domain master = Yes
  ldap suffix = ou=People,o=oss.ilabs.interop.net
  ldap machine suffix = ou=Computers,o=oss.ilabs.interop.net
  ldap user suffix = ou=users,o=oss.ilabs.interop.net
  ldap group suffix = ou=filegroups,o=oss.ilabs.interop.net
  ldap idmap suffix = ou=Idmap,o=oss.ilabs.interop.net
  ldap admin dn = cn=manager,o=oss.ilabs.interop.net
  ldap ssl = no
  idmap uid = 15000-20000
  idmap gid = 15000-20000

[netlogon]
  path = /etc/samba/netlogon
  writable = no

[homes]
  comment = Home Directories
  browseable = no
  writable = yes

[documents]
  comment = Document Directory
  path = /usr/local/samba
  browseable = yes
  writable = yes
```

Figure 6 /etc/smb.conf

```
[root@localhost ~]# more /etc/ldap.conf
host 45.230.3.10
base cn=Users,o=oss.ilabs.interop.net
binddn cn=ldap-user,cn=Users, dc=b,dc=oss,dc=ilabs,dc=interop,dc=net
bindpw ...
scope sub
ssl no
nss_base_passwd cn=Users,dc=b,dc=oss,dc=ilabs,dc=interop,dc=net
nss_base_shadow cn=Users,dc=b,dc=oss,dc=ilabs,dc=interop,dc=net
nss_base_group cn=Users,dc=b,dc=oss,dc=ilabs,dc=interop,dc=net
nss_map_objectclass posixAccount user
nss_map_objectclass shadowAccount user
nss_map_attribute uid sAMAccountName
nss_map_attribute uidNumber msSFU30UidNumber
nss_map_attribute gidNumber msSFU30GidNumber
nss_map_attribute loginShell msSFU30LoginShell
nss_map_attribute gecocnname
nss_map_attribute userPassword msSFU30Password
nss_map_attribute homeDirectory msSFU30HomeDirectory
nss_map_objectclass posixGroup Group
nss_map_attribute uniqueMember msSFU30PosixMember
nss_map_attribute cn cn
pam_login_attribute sAMAccountName
pam_filter objectclass=user
pam_member_attribute msSFU30PosixMember
pam password crypt
```

Figure 7 /etc/ldap.conf

Linux authentication against Active Directory

Getting Linux to authenticate against Active Directory was a whole different challenge. This section describes how we made the Linux client on site D authenticate against the AD server. Again there are several good documents describing parts of this procedure available via a quick web search.

We installed Microsoft's Services for UNIX, choosing the defaults all the way where possible and answering a couple of simple questions. The whole installation was straight forward. Furthermore, we created a user in Active Directory for the Linux client to use when binding to Active Directory.

Finally we added the POSIX attributes to the groups and users in Active Directory. Manually this can be done as follows. Choose a group and modify it, choose the UNIX attributes tab and modify GID etc. as appropriate. Choose the user you want to make POSIX compliant, modify it, choose the Unix attributes tab and add UID, GID, home directory etc. as appropriate. Add the user to the UNIX group. Reset the user's password to synchronize the Active Directory and UNIX passwords. In our setup, this was all done through cerebrum.

On the Linux client, we ran `authconfig` (as described above) to enable PAM to use LDAP for authentication. We made sure the Server field was set to our domain controller, not to select TLS and set or BaseDN to "cn=users,o=oss.ilabs.net". We selected "Use LDAP authentication", "Use Shadow Passwords" and "Use MD5 Passwords". This took care of the `/etc/nsswitch.conf` and `/etc/pam.d/system-auth` files, but we had to manually edit the `/etc/ldap.conf` file as shown in **figure 7**.

LDIFs

Figure 8 shows the Ldif for the people that is passed to OpenLDAP. **Figure 9** shows the Ldif for the users tree that is passed to OpenLDAP. A script on the OpenLDAP server does a `ldapmodify`. There is no Ldif passed to AD, but rather we update the LDAP info in AD directly through an ADSI.

```
dn: uid=janed,ou=users,o=oss.ilabs.interop.net
cn: Jane Doe
gecos: JaneDoe
gidNumber: 2000
homeDirectory: /home/janed
loginShell: /bin/bash
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
objectClass: sambaSamAccount
sambaAcctFlags: [UX]
sambaHomeDrive: M:
sambaHomePath: \\SAMBA\homes
sambaLogonScript: logon.bat
sambaNTPassword:
B50E56BB624C8E9778D7F6C2B4FCFE5D
sambaProfilePath: \\SAMBA\homes\profiles
sambaPwdCanChange: 0
sambaPwdLastSet: 1
sambaSID: S-1-5-21-3404150678-3940928994-
1262612553-3012
uid: janed
uidNumber: 1001
userPassword: alsdjf;lskdjfk
```

```
dn: uid=janed,ou=People,o=oss.ilabs.interop.net
cn: Jane Doe
eduPersonOrgDN: o=oss.ilabs.interop.net
eduPersonOrgUnitDN:
ou=D,ou=organization,o=oss.ilabs.interop.net
eduPersonPrimaryOrgUnitDN:
ou=D,ou=organization,o=oss.ilabs.interop.net
givenName: Jane
mail: jane.doe@d.oss.ilabs.interop.net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: eduPerson
sn: Doe
uid: janed
userPassword: $1$1Q2wKCq0$DyNOwPCOsBwuU6P7wibv01
```

Figure 8 People

Figure 9 Users

Reference

Cerebrum: <http://www.cerebrum.uio.no> or <http://cerebrum..sourceforge.net>

OpenLDAP: <http://www.openldap.org>

Samba: <http://www.samba.org>

Kerberos: <http://www.mcmce.com/win2k/guides/kerberos.shtml#win2k>