Using Kerberos to Authenticate a Solaris™ 10 OS LDAP Client With Microsoft Active Directory

Wajih Ahmed and Baban Kenkre

March 2008 (Updated May 2008)

Sun Microsystems, Inc.

Please note: *This configuration uses a shell script called* `adjoin.sh` *to automate the process of joining the Solaris client to the Active Directory domain and configures Kerberos on the client. This script is not supported by Sun and is not part of the Solaris distribution. (See the For More Information section for information about downloading the* `adjoin` *script.)*

THE SOLUTION DESCRIBED IN THIS PAPER SHOULD BE TREATED AS PROOF OF CONCEPT AND SHOULD NOT BE USED IN PRODUCTION.

Table of Contents

Introduction

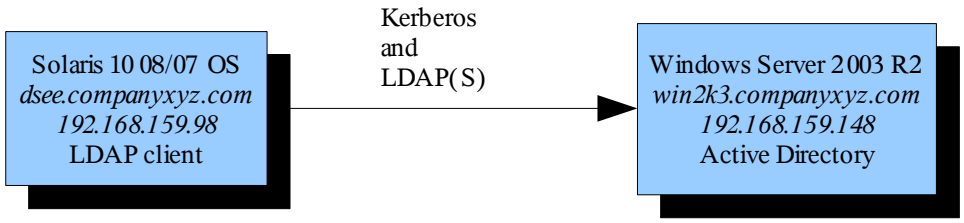
This document describes how to configure a Solaris Operating System client to use Microsoft Windows Server 2003 R2 Enterprise Edition (Active Directory) for authentication and naming services. The Solaris client uses per-user authentication (also called self-credentials) for naming service lookups instead of a proxy account. This new functionality is available starting with the Solaris 10 08/07 OS.

This configuration uses a shell script called `adjoin.sh` to automate the process of joining the Solaris client to the Active Directory domain and configures Kerberos on the client. This script is not supported by Sun and is not part of the Solaris distribution. (See the For More Information section for information about downloading the `adjoin` script.)

THE SOLUTION DESCRIBED IN THIS PAPER SHOULD BE TREATED AS PROOF OF CONCEPT AND SHOULD NOT BE USED IN PRODUCTION.

Figure 1 illustrates the example topology used in this document.

Figure 1: Example Topology



The remaining sections of this document describe how to perform the configuration steps on the Microsoft Windows and Solaris systems.

Note: This document does not cover the installation of the operating systems or the Active Directory service.

Perform the following tasks on the Microsoft Windows system:

- Install Microsoft Windows Server 2003 R2 Enterprise Edition.
- Configure the Microsoft Windows server as a domain controller with “typical” options and a static IP address.
- Install the Active Directory service.
- Install Identity Management for UNIX®.
- Add Domain Name System (DNS) records for the Solaris client.
- Optionally, tune Active Directory.
- Add or provision an Active Directory test user with UNIX attributes.

Perform the following tasks on the Solaris system:

- Install at least the Solaris 10 08/07 release and ensure that the Kerberos client packages are installed.
- Download and run the `adjoin` tool.
- Run `ldapsearch` with the Generic Security Services Application Programming Interface (GSSAPI) mechanism to test connectivity.

- Initialize the Solaris LDAP client.
- Test the LDAP client.

Installing Identity Management for UNIX

After installing Active Directory on the Microsoft Windows server, perform the following steps to support the POSIX schema in Active Directory. On Microsoft Windows Server 2003 R2 Enterprise Edition, you must install Identity Management for UNIX. If you are using a prior version of the Microsoft Windows server, install Services for UNIX (SFU), which can be downloaded from the Microsoft web site.

Note: Before installing Identity Management for UNIX on Microsoft Windows Server 2003 Enterprise Edition, uninstall SFU if it was previously installed.

1. In the control panel, choose Add or Remove Programs.
2. Click Add/Remove Windows Components.
3. Select Active Directory Services.
4. Click Details.
5. Select only Identity Management for UNIX, as shown in Figures 2a, b, and c.

Figure 2a: Installing Identity Management for UNIX - Initiate

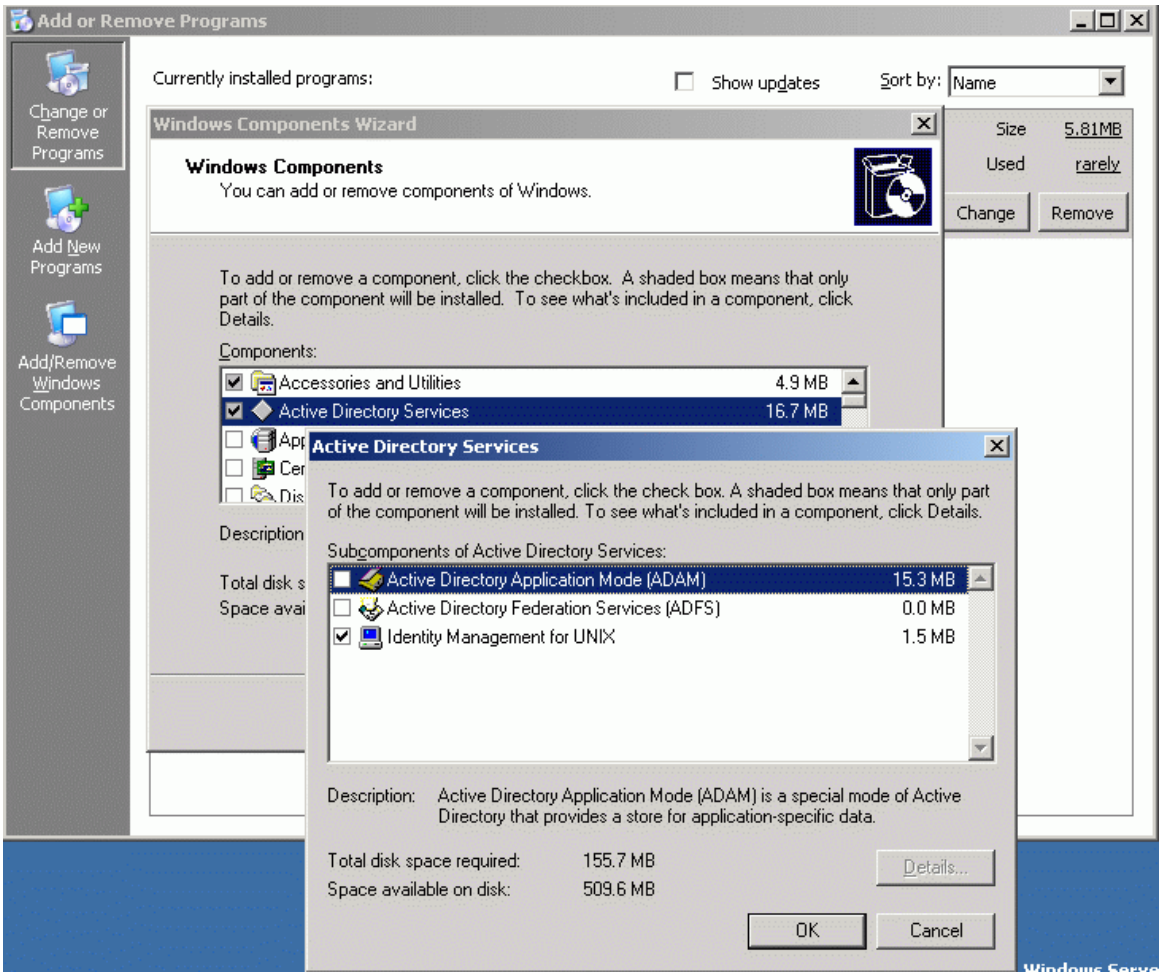


Figure 2b: Installing Identity Management for UNIX - Progress

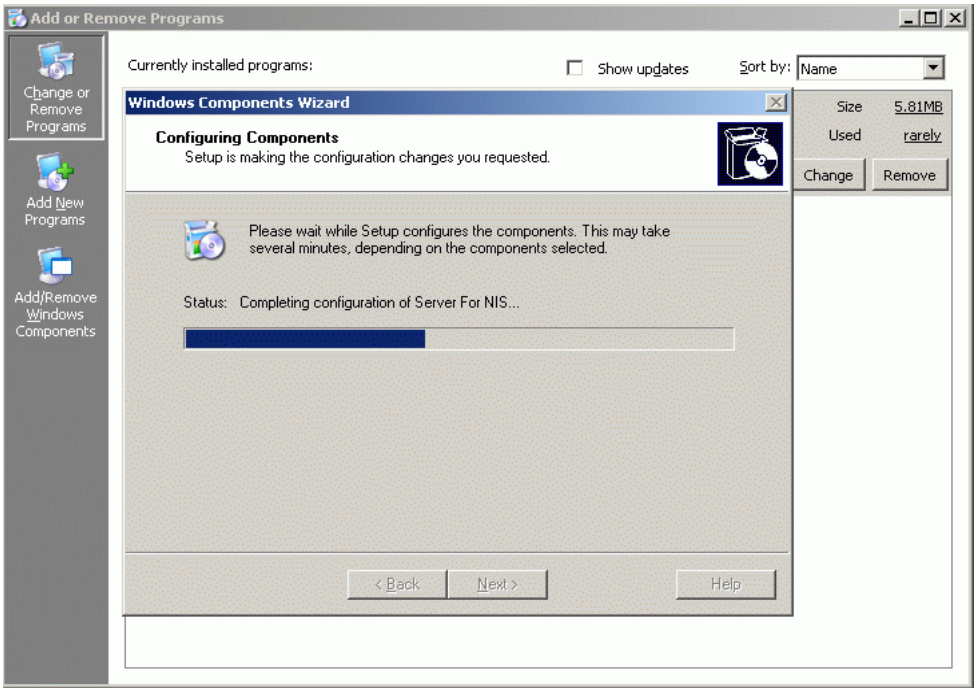


Figure 2c: Installing Identity Management for UNIX - Complete

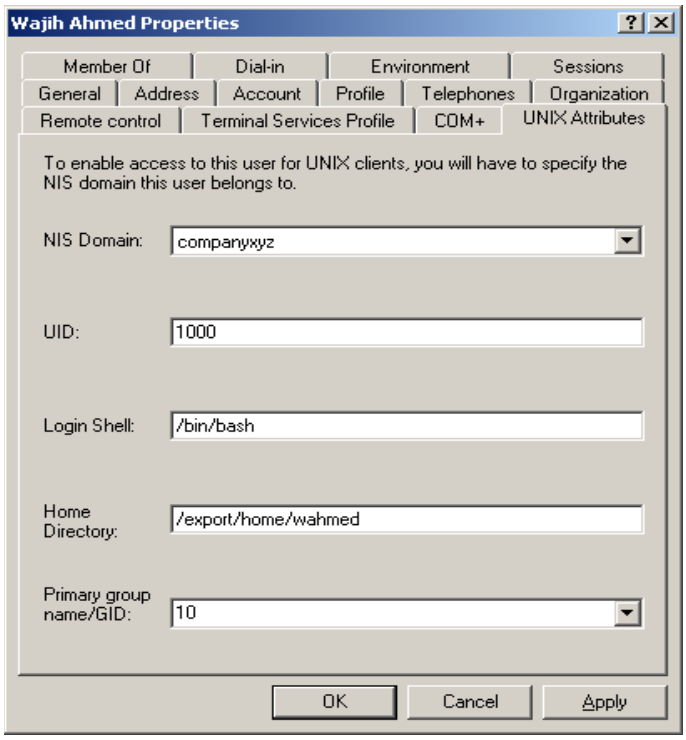
After the installation is complete, reboot the Microsoft Windows server.

Provisioning a UNIX User in Active Directory

The next step is to add UNIX attributes to Active Directory users and groups on the Microsoft Windows system. Identity Management for UNIX adds the UNIX Attributes tab to the user's and group's Properties page for this purpose. This tab also appears if you are using SFU.

The new Active Directory user, `wahmed`, is used to test the configuration by resetting its password and populating its UNIX attributes. While the NIS domain shown in Figure 3 appears without the fully qualified domain name (`companyxyz`), the domain used by the Solaris client is fully qualified (`companyxyz.com`).

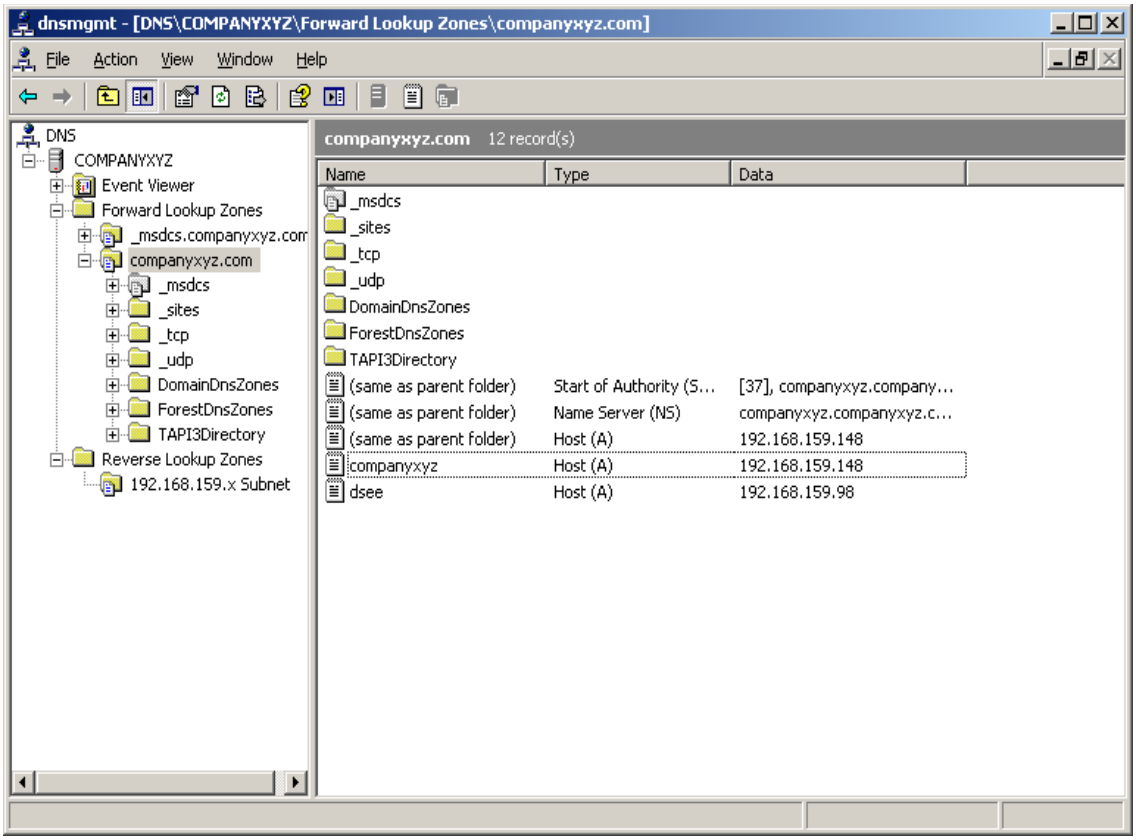
Figure 3: NIS Domain



Configuring DNS

On the Microsoft Windows system, create a forward (A) and reverse (PTR) DNS record for the Solaris client. In addition, create a reverse (PTR) DNS record for the AD server. These records are required for Kerberos to function properly. The forward (A) DNS record for the Active Directory server is created automatically when configuring the Active Directory server. The following example assumes that you are using Active Directory as the DNS server.

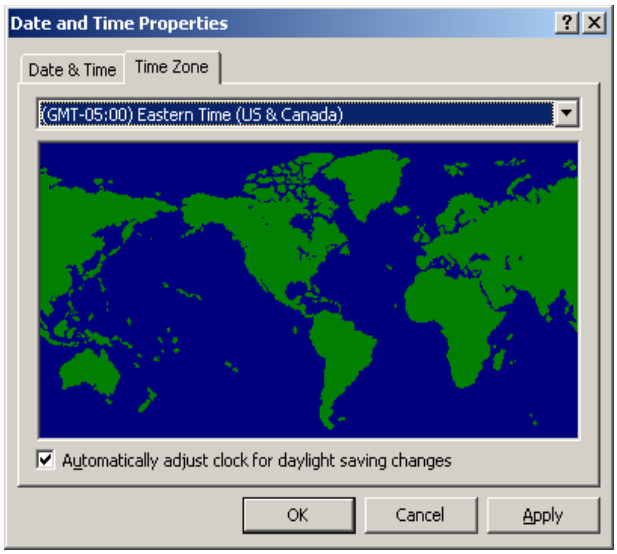
Figure 4: Forward and Reverse Lookup Zones



Synchronizing the Clocks and Configuring Time Zones

Time synchronization is *essential* for Kerberos to function properly. By default, only a 300-second clock skew is acceptable. Ensure that time zones on all Microsoft Windows and Solaris servers are configured properly. You can use NTP to synchronize time.

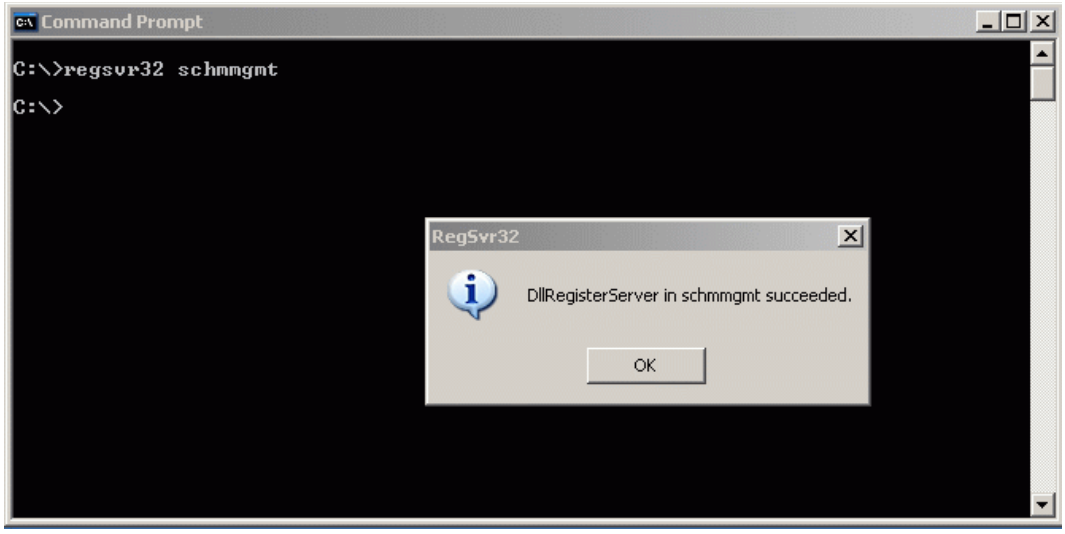
Figure 5: Configuring Time Zones



Tuning Active Directory

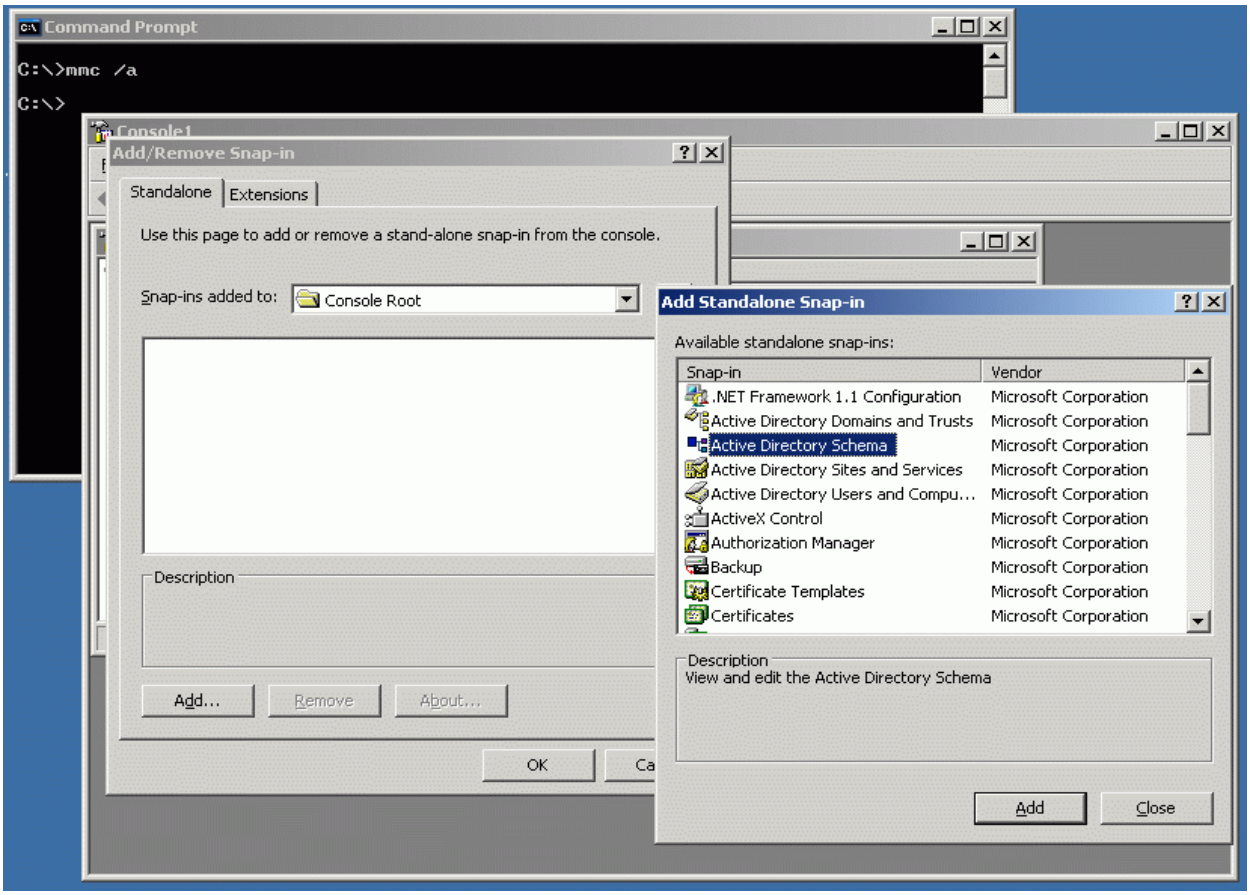
On the Microsoft Windows system, index the following Solaris client attributes: `uid`, `uidnumber`, `gid`, and `gidnumber`. In Active Directory, indexes can be added by using the Schema Management Snap-In for the Microsoft Management Console. This snap-in must be registered first, as shown in Figure 6.

Figure 6: Registering the Schema Management Snap-In



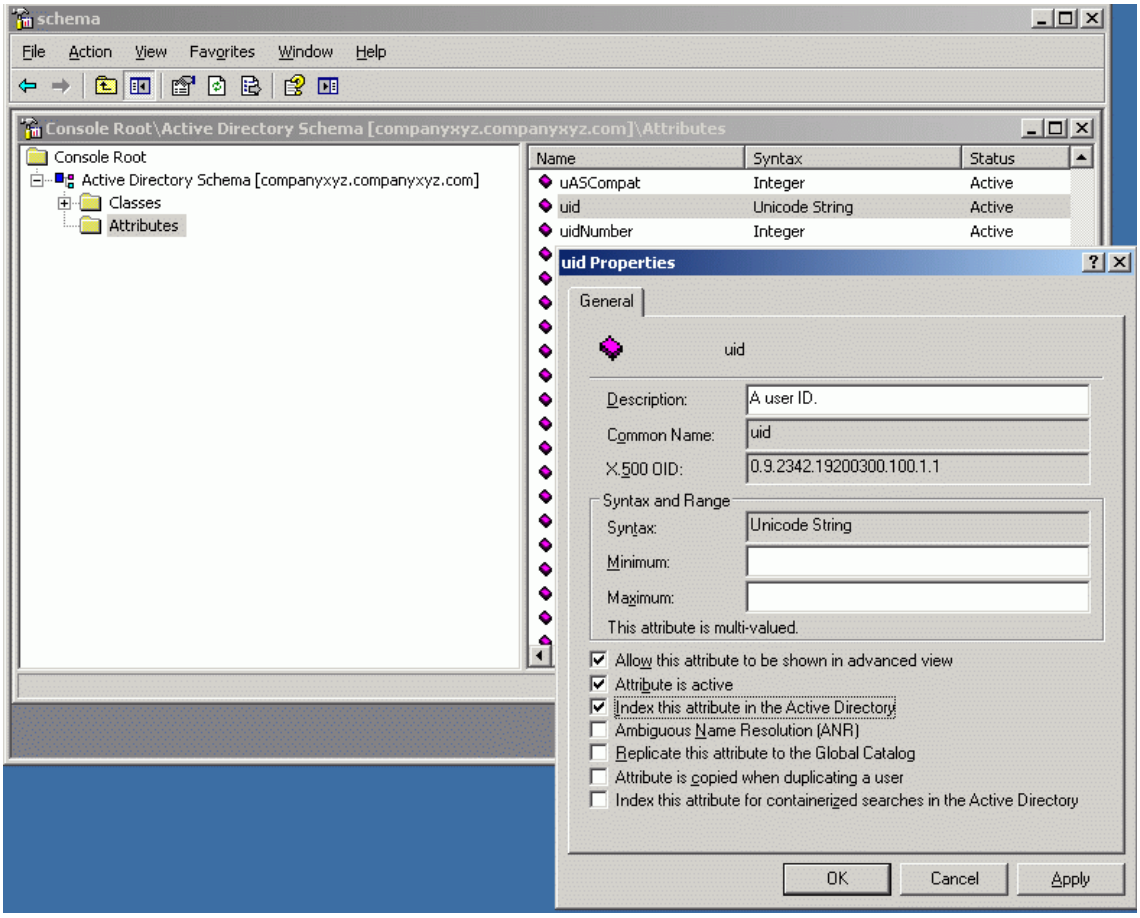
Then, add the Active Directory Schema plug-in to the Microsoft Management Console by running `mmc /a` from the command prompt, as shown in Figure 7.

Figure 7: Adding the Directory Schema Plug-In



Save the snap-in to a file, such as `schema.msc`, for later use. Figure 8 shows how to index the `uid` attribute.

Figure 8: Indexing the `uid` Attribute



Configuring Kerberos

The Solaris client must join an Active Directory domain to use Active Directory for security and directory services. The `adjoin.sh` script automates the domain join operation by executing the following steps from the Solaris client:

- Auto-detects the Active Directory domain controller
- Creates a machine account (also called a Computer object) for the Solaris host in Active Directory and generates a random password for this account
- Configures the Solaris host as a Kerberos client of the Active Directory domain controller by using the `/etc/krb5/krb5.conf` file
- Configures the `/etc/krb5/krb5.keytab` file on the Solaris host by using the keys for the machine account (also called host credentials)

The `adjoin.sh` script uses the `ksetpw` binary to set the password for the machine account and to configure the local `keytab` file. Run `adjoin -h` to see the options supported by the `adjoin.sh` script. This script requires proper DNS configuration on the client. Therefore, `/etc/resolv.conf` must point to the correct DNS domain and servers, and `/etc/nsswitch.conf` must use DNS for host resolution. Ensure that the `ksetpw` binary is in the same directory as `adjoin.sh`.

In the following example, the Solaris client is using the Active Directory server as its DNS server.

```
dsee% cat /etc/resolv.conf
```

```
domain companyxyz.com
nameserver 192.168.159.148
```

```
dsee% egrep "hosts|ipnodes" /etc/nsswitch.conf
```

```
hosts: files dns
ipnodes: files dns
```

The following `adjoin.sh` example output is for a Solaris host, `dsee`, that tries to join an Active Directory domain, `companyxyz.com`, that is served by the Active Directory domain controller, `win2k3`. The `-f` option forces the creation of a machine account for `dsee` even if one already exists. If a machine account already exists, the existing account is first removed before being recreated.

```
dsee% ./adjoin.sh -f
```

```
Joining domain: companyxyz.com
Looking for domain controllers and global catalogs (A RRs)
Looking for KDCs and DCs (SRV RRs)
        KDCs = win2k3.companyxyz.com 88
        DCs = wins2k3.companyxyz.com 389
Password for Administrator@COMPANYXYZ.COM:
Looking for forest name
        Forest name = companyxyz.com
Looking for Global Catalog servers
Looking for site name
        Looking for subnet object in the global catalog
Could not find site name for any local subnet
        Site name not found.  Local DCs/GCs will not be discovered
Looking to see if there's an existing account...
Looking to see if the machine account contains other objects...
Deleting existing machine account...
Creating the machine account in AD via LDAP
adding new entry CN=DSEE,CN=Computers,DC=companyxyz,DC=com

Setting the password/keys of the machine account
Result: success (0)
Getting kvno
KVNO: 2
Determining supported enctypes for machine account via LDAP
This must not be a Longhorn/Vista AD DC!
        So we assume 1DES and arcfour enctypes
ARCFOUR will be supported
Finishing machine account
modifying entry CN=DSEE,CN=Computers,DC=companyxyz,DC=com

adjoin: Done
```

Verify the setup by running the following `ldapsearch` command to bind to Active Directory by using GSSAPI. The command uses the host credentials that have been created for the Solaris client by the `adjoin.sh` script. If the command runs without any errors, the setup is correct. It does not matter which base (`-b`) is used for the following command:

```
dsee% ldapsearch -h win2k3.companyxyz.com -o mech=gssapi -o authzid='' -b
"cn=dsee,cn=computers,dc=companyxyz,dc=com" -s base "" cn
```

```
version: 1
```

```
dn: cn=dsee,cn=computers,dc=companyxyz,dc=com
cn: DSEE
```

Use `klist` to display the the Kerberos ticket cache.

```
dsee% klist
```

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: host/dsee.companyxyz.com@COMPANYXYZ.COM
Valid starting                Expires                Service principal
10/27/07 10:52:05  10/27/07 20:51:20  krbtgt/COMPANYXYZ.COM@COMPANYXYZ.COM
        renew until 10/27/07 20:52:05
10/27/07 10:51:20  10/27/07 20:51:20  ldap/win2k3.companyxyz.com@COMPANYXYZ.COM
        renew until 10/27/07 20:52:05
```

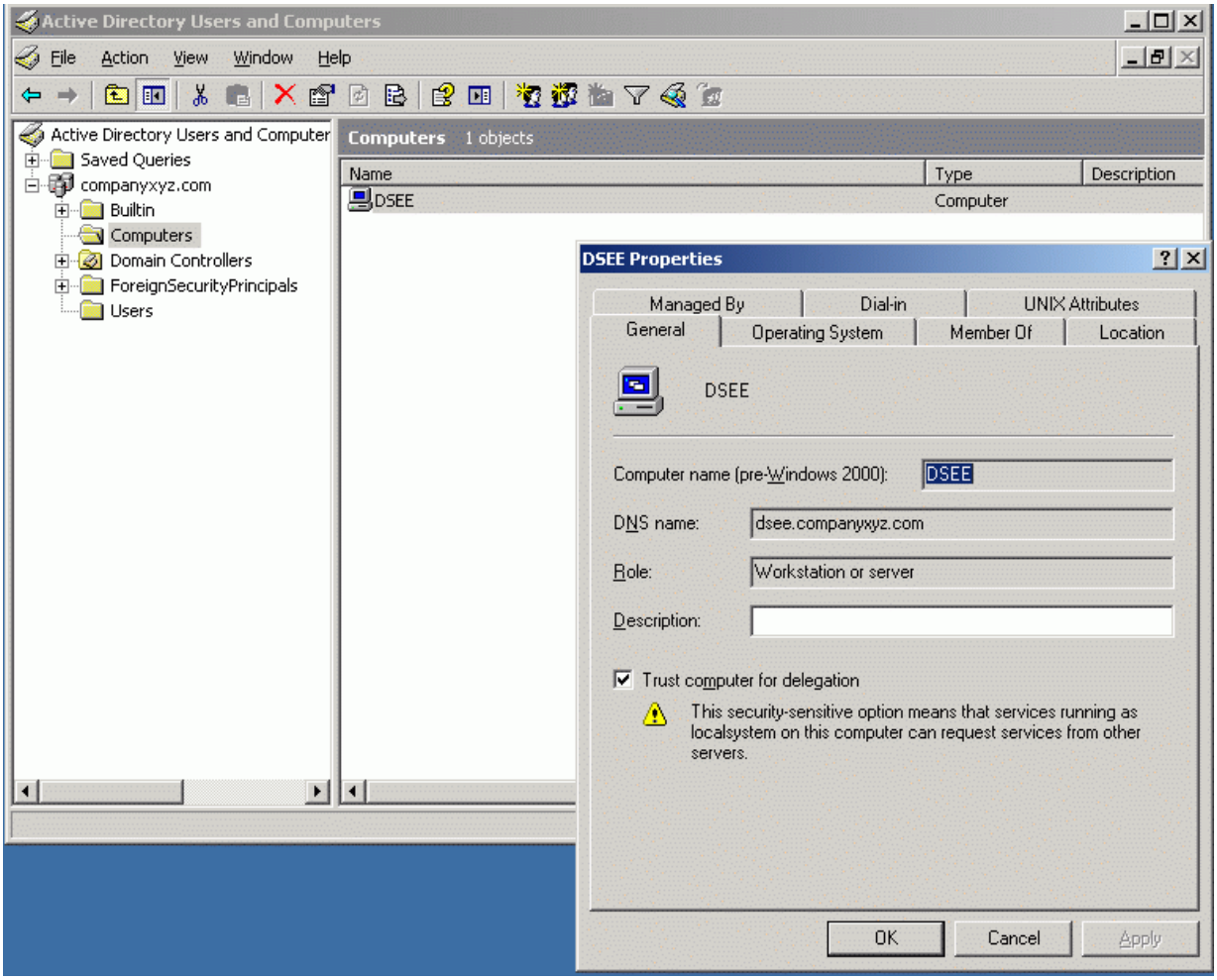
List the host keys for the Solaris client `dsee` by running the following `klist` command:

```
dsee% klist -e -k /etc/krb5/krb5.keytab
```

```
Keytab name: FILE:/etc/krb5/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 host/dsee.companyxyz.com@COMPANYXYZ.COM (ArcFour with HMAC/md5)
   1 host/dsee.companyxyz.com@COMPANYXYZ.COM (DES cbc mode with CRC-32)
   1 host/dsee.companyxyz.com@COMPANYXYZ.COM (DES cbc mode with RSA-MD5)
```

The Active Directory console now shows an entry for Solaris client (DSEE) under the Computers container, as shown in Figure 9.

Figure 9: Entry for Solaris Client



The contents of the `krb5.conf` file should be as follows:

```
dsee% cat /etc/krb5/krb5.conf
```

```
[libdefaults]
        default_realm = COMPANYXYZ.COM

[realms]
        COMPANYXYZ.COM = {
                kdc = win2k3.companyxyz.com
                kpasswd_server = win2k3.companyxyz.com
                kpasswd_protocol = SET_CHANGE
                admin_server = win2k3.companyxyz.com
        }

[domain_realm]
        .companyxyz.com = COMPANYXYZ.COM
```

```
dsee% ldapsearch  -h win2k3.companyxyz.com -b "cn=users,dc=companyxyz,dc=com" -o
mech=gssapi -o authzid='' "cn=wajih ahmed"
```

Use the `ldapsearch` command for a user to ensure the presence of POSIX attributes. In the following output, the attributes in **bold** were added by Identity Management for UNIX and those in *italics* are the SFU attributes. Actual output only shows one set of attributes. The example shows both sets to highlight the attribute names.

```
version: 1
dn: CN=Wajih Ahmed,CN=Users,DC=companyxyz,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Wajih Ahmed
sn: Ahmed
givenName: Wajih
initials: U
distinguishedName: CN=Wajih Ahmed,CN=Users,DC=companyxyz,DC=com
instanceType: 4
whenCreated: 20071023182249.0Z
whenChanged: 20071030162832.0Z
displayName: Wajih Ahmed
uSNCreated: 16413
uSNChanged: 69721
name: Wajih Ahmed
objectGUID:: hJdxx1sW3EeLoIHl+nZjKQ==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 128381558123360000
lastLogoff: 0
lastLogon: 128382298039330000
pwdLastSet: 128381558243047500
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAA72Kb5qzJ+t5uyLN+VQQAAA==
accountExpires: 9223372036854775807
logonCount: 15
sAMAccountName: wahmed
sAMAccountType: 805306368
userPrincipalName: wahmed@companyxyz.com
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=companyxyz,DC=com
uid: wahmed
uidNumber: 1000
gidNumber: 10
unixHomeDirectory: /export/home/wahmed
loginShell: /bin/bash
msSFU30Name: wahmed
msSFU30UidNumber: 1000
msSFU30GidNumber: 10
msSFU30LoginShell: /bin/bash
msSFU30Password: uTtVgSFk3So22
msSFU30NisDomain: companyxyz
msSFU30HomeDirectory: /export/home/wahmed
```

Initializing the Solaris LDAP Client

Now you configure the Solaris host as an LDAP client of Active Directory, which allows the Solaris host to access naming service information from Active Directory.

As prerequisites, the DNS client and `nscd` should be enabled, and the `/etc/resolv.conf` file should be properly configured. Verify that both forward and reverse DNS lookup of the Active Directory server succeeds from the Solaris host, as shown in the following example.

If reverse DNS lookup fails, then add a PTR record for the Active Directory server to the DNS server, if it does not exist. Modify `/etc/nsswitch.ldap` to use DNS for hosts and ipnodes. Unlike earlier versions, `nscd` in the Solaris 10 08/07 release supports enhanced LDAP connection management and improved caching. You must enable `nscd` to use the per-user authentication functionality as follows:

```
dsee% svcadm enable svc:/network/dns/client:default
dsee% svcadm enable name-service-cache
dsee% dig win2k3.companyxyz.com +short
192.168.159.148
dsee% dig -x 192.168.159.148 +short
win2k3.companyxyz.com.
dsee% grep dns /etc/nsswitch.ldap
hosts:   dns files
ipnodes: dns files
```

In the following example, Microsoft Windows Server 2003 Enterprise Edition R2 has Identity Management for UNIX enabled. The POSIX attributes and object classes added to the Active Directory schema by Identity Management for UNIX have the same names as those used by the Solaris LDAP client (which follows the RFC2307bis IETF draft) except for those that must use attribute mapping (`attributeMap`) and object class mapping (`objectClassMap`).

```
dsee% ldapclient -v manual \
-a credentialLevel=self \
-a authenticationMethod=sasl/gssapi \
-a defaultSearchBase=dc=companyxyz,dc=com \
-a domainName=companyxyz.com \
-a defaultServerList=192.168.159.148 \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:cn=users,dc=companyxyz,dc=com?one \
-a serviceSearchDescriptor=group:cn=users,dc=companyxyz,dc=com?one
```

The use of `credentialLevel=self` denotes per-user authentication, which means that the Solaris LDAP client uses the credentials of the user who is making the naming service request to bind and look up information in the LDAP server (Active Directory, in this case).

The use of `authenticationMethod=sasl/gssapi` denotes that the Solaris LDAP client uses GSSAPI/Kerberos to authenticate to the LDAP server. The per-user authentication can be used only in conjunction with `sasl/gssapi`. (SASL refers to Simple Authentication and Security Layer.)

The following example uses an older version of Microsoft Windows Server 2003, which has SFU installed. This configuration needs additional attribute mappings.

```
dsee% ldapclient -v manual \
-a credentialLevel=self \
-a authenticationMethod=sasl/gssapi \
-a defaultSearchBase=dc=companyxyz,dc=com \
-a domainName=companyxyz.com \
-a defaultServerList=192.168.159.148 \
-a attributeMap=group:userpassword=msSFU30Password \
-a attributeMap=group:memberuid=msSFU30MemberUid \
-a attributeMap=group:gidnumber=msSFU30GidNumber \
-a attributeMap=passwd:gecos=msSFU30Gecos \
-a attributeMap=passwd:gidnumber=msSFU30GidNumber \
-a attributeMap=passwd:uidnumber=msSFU30UidNumber \
-a attributeMap=passwd:uid=sAMAccountName \
-a attributeMap=passwd:homedirectory=msSFU30HomeDirectory \
-a attributeMap=passwd:loginshell=msSFU30LoginShell \
-a attributeMap=shadow:shadowflag=msSFU30ShadowFlag \
-a attributeMap=shadow:userpassword=msSFU30Password \
-a attributeMap=shadow:uid=sAMAccountName \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:cn=users,DC=companyxyz,DC=com?one \
-a serviceSearchDescriptor=group:cn=users,DC=companyxyz,DC=com?one
```

You should see the `Successfully configured` message after running the `ldapclient` command. Restart the LDAP client.

```
dsee% svcadm restart svc:/network/ldap/client:default
```

Verify the contents of the LDAP client cache. The output is different if you are using SFU.

```
dsee% ldapclient list
```

```
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_SERVERS= 192.168.159.148
NS_LDAP_SEARCH_BASEDN= dc=companyxyz,dc=com
NS_LDAP_AUTH= sasl/GSSAPI
NS_LDAP_CREDENTIAL_LEVEL= self
NS_LDAP_SERVICE_SEARCH_DESC= passwd:cn=users,dc=companyxyz,dc=com?one
NS_LDAP_SERVICE_SEARCH_DESC= group:cn=users,dc=companyxyz,dc=com?one
NS_LDAP_ATTRIBUTEMAP= passwd:homedirectory=unixHomeDirectory
NS_LDAP_ATTRIBUTEMAP= passwd:gecos=cn
NS_LDAP_OBJECTCLASSMAP= shadow:shadowAccount=user
NS_LDAP_OBJECTCLASSMAP= passwd:posixAccount=user
```

```
NS_LDAP_OBJECTCLASSMAP= group:posixGroup=group
```

Using the Naming Service Switch and Pluggable Authentication Modules (PAM)

The following `/etc/nsswitch.conf` file configures the Solaris client to use Active Directory for users and groups, DNS for host resolution, and local files for other naming service lookups:

```
dsee% cat /etc/nsswitch.conf
```

```
passwd:     files ldap
group:      files ldap
hosts:      dns files
ipnodes:    dns files
networks:   files
protocols:  files
rpc:        files
ethers:     files
netmasks:   files
bootparams: files
publickey:  files
# At present there isn't a 'files' backend for netgroup;  the system will
#   figure it out pretty quickly, and won't use netgroups at all.
netgroup:   files
automount:  files
aliases:    files
services:   files
printers:   user files
auth_attr:  files
prof_attr:  files
project:    files
tnrhtp:     files
tnrhdb:     files
```

Use the `pam_krb5.so.1` module in the `/etc/pam.conf` file to enable authentication, account management, and password management on the Solaris client by using Active Directory through Kerberos. Minimally, enable the module for `login` and `other` services.

The following `/etc/pam.conf` file authenticates users by using Active Directory through Kerberos and authenticates through the UNIX login only if the Kerberos authentication fails (see the `auth` entries). This arrangement is helpful when a majority of the users are in Active Directory and when there are only a few non-Active Directory user accounts, such as `root`. The `account` entries check for password expiration when dealing with Active Directory and local UNIX password-aging policies. The `password` entries change the Active Directory password of the user and continue to change the local UNIX password only if the Active Directory password change fails.

```
login   auth requisite          pam_authtok_get.so.1
login   auth required           pam_dhkeys.so.1
login   auth required           pam_unix_cred.so.1
login   auth sufficient         pam_krb5.so.1
login   auth required           pam_unix_auth.so.1
login   auth required           pam_dial_auth.so.1
```

```
other   auth requisite          pam_authtok_get.so.1
other   auth required           pam_dhkeys.so.1
other   auth required           pam_unix_cred.so.1
other   auth sufficient         pam_krb5.so.1
other   auth required           pam_unix_auth.so.1

other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account required        pam_krb5.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password sufficient     pam_krb5.so.1
other   password required       pam_authtok_store.so.1
```

Testing the Client

Test the configuration by running the `getent` command for the `passwd` database for a particular user. If this command does not return the user, the client configuration failed. Check the `/var/adm/messages` file or the console for errors.

```
dsee$ getent passwd wahmed
```

```
wahmed:x:1000:10::/export/home/wahmed:/bin/bash
```

Use the `ldaplist` command to search for and list naming information.

Note that running the `ldaplist -l` command returns a `Critical Extension not found` error, but if you specify an Active Directory user, you should get the correct output. The critical extension error occurs because Active Directory does not support some of the LDAP Version 3 extensions that are used by the Solaris LDAP client. In particular, Active Directory does not support the extension that is required for virtual list view (VLV) indexes.

```
dsee$ ldaplist -l passwd wahmed
```

```
dn: gecos=Wajih Ahmed,gecos=Users,DC=companyxyz,DC=com
        objectClass: top
        objectClass: person
        objectClass: organizationalPerson
        objectClass: posixAccount
        cn: Wajih Ahmed
        sn: Ahmed
        givenName: Wajih
        initials: U
        distinguishedName: CN=Wajih Ahmed,CN=Users,DC=companyxyz,DC=com
        instanceType: 4
        whenCreated: 20071023182249.0Z
        whenChanged: 20071030162832.0Z
        displayName: Wajih Ahmed
        uSNCreated: 16413
        uSNChanged: 69721
        name: Wajih Ahmed
        objectGUID:   q [# G
        userAccountControl: 66048
        badPwdCount: 0
        codePage: 0
```

```
        countryCode: 0
        badPasswordTime: 128381558123360000
        lastLogoff: 0
        lastLogon: 128382298039330000
        pwdLastSet: 128381558243047500
        primaryGroupID: 513
        objectSid: #
        accountExpires: 9223372036854775807
        logonCount: 15
        sAMAccountName: wahmed
        sAMAccountType: 805306368
        userPrincipalName: wahmed@companyxyz.com
        objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=companyxyz,
         DC=com
        uid: wahmed
        msSFU30Name: wahmed
        msSFU30UidNumber: 1000
        msSFU30GidNumber: 10
        msSFU30LoginShell: /bin/bash
        msSFU30Password: uTtVgSFk3So22
        msSFU30NisDomain: companyxyz
        msSFU30HomeDirectory: /export/home/wahmed
        uidnumber: 1000
        gidnumber: 10
        homedirectory: /export/home/wahmed
        loginshell: /bin/bash
        gecos: Wajih Ahmed
```

Note: The `objectGUID` and `objectSID` attributes in the `ldaplist` output have binary values.

Verify that you can log in successfully to the Solaris client as an Active Directory user by using `ssh`. The following example uses a manually created local home directory. Home directories that are shared by an NFS server can be automatically mounted at login time by configuring `automount`(1M) on the Solaris client.

Note that the Solaris 10 08/07 release does not support the automounting of remote home directories using `smbfs`. The `smbfs` functionality has been integrated into OpenSolaris build 84. See the "For More Information" section for a link to the OpenSolaris CIFS client project.

```
localhost:~> ssh -l wahmed dsee
```

```
Password:
Last login: Thu Nov  1 19:05:32 2007 from gateway
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
```

```
dsee$ id
```

```
uid=1000(wahmed) gid=10(staff)
```

```
dsee$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: wahmed@COMPANYXYZ.COM

Valid starting                Expires                Service principal
11/06/07 10:23:12  11/06/07 20:19:30  krbtgt/COMPANYXYZ.COM@COMPANYXYZ.COM
        renew until 11/13/07 10:23:12
```

Testing Password Management

Following is a list of tests used to check account and password management with Active Directory. The results show that most of the commonly occurring scenarios work quite well. The tests were performed using a Solaris 10 08/07 client.

Note: Use `kpasswd` instead of the `passwd` command to change an Active Directory user's password from the Solaris client.

1. Log in and change the password of an Active Directory user from the Solaris client.

Active Directory: Reset the password for user `wahmed` to `Admin1234`.
Solaris client: Log in successfully as `wahmed` with the password `Admin1234`.
Solaris client: As `wahmed`, successfully use `kpasswd` to change password to `Abcd1234`. Then, log out.
Solaris client: Log in successfully as `wahmed` using the new password `Abcd1234`.

Note: `kpasswd` uses the Active Directory password policy for password changes. If the new password does not meet the Active Directory policy, `kpasswd` issues the `kpasswd: Password change rejected` error.

2. Require the user to change the password during next login.

Active Directory: Reset the password for user `wahmed` to `Admin1234`.

Active Directory: Set `User must change password on next logon`.

Solaris client: Log in successfully as `wahmed` using the password `Admin1234`.

Solaris client: When prompted, enter the new password:

```
% ssh -l wahmed dsee
Password:
Your Kerberos password has expired.

New Password:
Re-enter new Password:
Kerberos password successfully changed

Last login: Tue Feb  5 16:44:14 2008 from somewhere-
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
dsee$ id
uid=1000(wahmed) gid=10(staff)
dsee$
```

Note: In this case, PAM is changing the user password. The `pam.conf` file being used ensures that the password meets both the Active Directory password policy and the local (`/etc/default/passwd`) policy. If the new password does not meet the local policy, the password change fails and the output includes the appropriate error message. For example, you might see the following error if the password is not long enough: `Password too short - must be at least 6 characters.` However, if the new password does not meet the Active Directory

password policy, the user sees only the `Password change rejected` error, not an explanation of the failure.

In the following example, the Active Directory password policy requires a password with a minimum of seven characters, while the local policy requires a minimum of six characters. If you type a new password with six characters, it is rejected by Active Directory.

```
% ssh -l wahmed dsee
Password:
Your Kerberos password has expired.

New Password:
Re-enter new Password:
Kerberos password not changed:
Password change rejected
```

3. Disable the account by using the Active Directory console.

Active Directory: Disable the `wahmed` account.

Solaris client: Log in as `wahmed` as follows:

```
% ssh -l wahmed dsee
Password:
Password:
Password:
Permission denied (gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive).
```

4. Check the password quality and strength.

Active Directory: Set the password policy to require a minimum of seven characters.

Solaris client: Enter a new 6-character password for user `wahmed` by using `kpasswd`.

```
dsee$ id
uid=1000(wahmed) gid=10(staff)
dsee$ kpasswd
kpasswd: Changing password for wahmed@COMPANYXYZ.COM.
Old password:
New password:
New password (again):
kpasswd: Password change rejected
```

Solaris client: Enter a new 7-character password for `wahmed` by using `kpasswd`.

```
dsee$ kpasswd
kpasswd: Changing password for wahmed@COMPANYXYZ.COM.
Old password:
New password:
New password (again):
Kerberos password changed.
dsee$
```

Troubleshooting

If Active Directory is down, and hence the key distribution center (KDC) is not responding, the Solaris LDAP client might go into maintenance mode. After the Active Directory server is up, you can "clear" the service.

```
dsee% svcs | grep ldap
maintenance    10:44:41 svc:/network/ldap/client:default
```

```
dsee% svcadm clear ldap/client
dsee% svcs | grep ldap
online         10:25:00 svc:/network/ldap/client:default
```

Note that login fails if the domain is not set correctly in the `/etc/resolv.conf` file or if `nscd` is not running. The `ldapclient` command fails if `/etc/nsswitch.ldap` has not been modified to use DNS for hosts and ipnodes or if reverse DNS lookup of the Active Directory server fails from the Solaris client.

For More Information

Download the `adjoin` tool:
http://opensolaris.org/os/project/winchester/files/adjoin-s10u4.tar.gz

A new version of adjoin tool is available for the Solaris 10 5/08 release:
http://opensolaris.org/os/project/winchester/files/adjoin-s10u5.tar.gz

This version contains an updated `ksetpw` source and binary which has been modified to run on the Solaris 10 5/08 OS. See README file for more details. Note that the `ksetpw.c` source file in this version can also be used on OpenSolaris systems.

Note: THE SOLUTION DESCRIBED IN THIS PAPER SHOULD BE TREATED AS PROOF OF CONCEPT AND SHOULD NOT BE USED IN PRODUCTION.

Here are additional resources:

- Training courses available at http://www.sun.com/training/:
  - Using LDAP as a Naming Service (IN-351)
  - Enterprise Security Using Kerberos and LDAP (SC-360)
  - LDAP Design and Deployment (WI-3501)
- Support:
  - Register your Sun gear: https://inventory.sun.com/inventory/
  - Services: http://www.sun.com/services
  - SunSolve Online: http://sunsolve.sun.com
- Open source resources:
  - Sparks project, which provides naming service enhancements: http://www.opensolaris.org/os/project/sparks/overview/
  - CIFS client for Solaris: http://www.opensolaris.org/os/project/smbfs/

- Related documents:
  - Sun BluePrints™ book *LDAP in the Solaris Operating Environment: Deploying Secure Directory Services*: http://www.sun.com/books/catalog/haines_bialaski_ldap.xml
  - Documentation at http://docs.sun.com

    Discussions, such as the Solaris Forums at http://forum.java.sun.com/index.jspa?tab=solaris
- Related web sites and articles:
  - Solaris Information Center on the BigAdmin web site: http://www.sun.com/bigadmin/hubs/documentation/
  - "Kerberos and LDAP Troubleshooting Tips" on Microsoft web site: http://www.microsoft.com/technet/solutionaccelerators/cits/interopmigration/unix/usecdirw/17wsdsu.mspx
- Events of interest to users of Sun products:
  - Worldwide developer events: http://developers.sun.com/events/
  - Current events: http://www.sun.com/events/index.jsp

Licensing Information

Acknowledgements

The authors would like to thank Sundeep Dhall and Cathleen Reiher for their help.

Change Log

May 2008 – Added disclaimer on cover page and information on Solaris 5/08 in For More Information section.