Using Active Directory as your Solaris Authentication Source

The scope of this paper is to document how a newly installed Solaris 10 server can be configured to use an Active Directory directory service as an authentication source. This is not a definitive guide, and is the result of the following request:

> *Scenario: A customer has a variety of "fat-client" Windows-based applications they wish to publish to multiple client types through the use of a kind of portal. They will also be introducing some new Solaris/Linux-based X-Windows applications as low-cost alternatives to Windows-based applications. The customer will use SGD to publish these applications through their custom portal, and have an existing Active Directory directory service to act as their authentication source. They will configure SGD to use AD as their login authority; how can they configure Solaris/Linux to use the same authentication information?*

To be clear, this setup uses AD as the authentication source, and all account information is to be maintained in AD – there is no replication of user data between AD and the Solaris clients; there are other ways to achieve what can be viewed to be as similar results, such as:

- Account Replication / Synchronization – by using password / account synchronization tools, (used by many Identity Management tools, including the "Server for NIS" installed as part of this process), accounts are replicated /synchronized among the various authentication sources, and are "linked" together by some common attribute;
- Single Sign-On (SSO) techniques, such as used by SGD, in which SGD "learns" user credentials for each different system, remembering them for future use, but with no replication or synchronization provided amongst the different platforms. And, of course, the user (or perhaps administrator) has to manually provide these different credentials, at least once.

The things you need to accomplish include:

- Extend the Active Directory schema to include "Unix" operating system attributes
- Establish the Solaris server as a participant in the Kerberos realm
- Setup Solaris' ldap client to map AD attributes to Unix attributes
- Setup the Solaris PAM configuration to use Kerberos/LDAP (Active Directory)

In the following, I'm using Server 2003 R2 ("R2" installs as an add-on to Server 2003) – this adds a number of interoperability features to Server 2003; for our purposes, the

addition of the "Services For Unix" product to the base operating system is the most important, as it extends the AD schema to include Unix attributes, such as uid and gid, that aren't present in the basic AD schema. RFC 2307 is the manner that Microsoft (and Vintela's and Centrify's) chose to provide these attributes; it shouldn't be assumed, however, that these three products are interoperable, or chose to implement the RFC in exactly the same way.

Step-By-Step Instructions

1. **Environmental Setup**

   Time Service – Kerberos is dependent on time being synchronized between the client and server; too large a clock skew will cause authentication failures. To prevent this, each server should use the same time source. By default, the ntp service isn't started on Solaris. To setup a Solaris 10 server as an ntp client, (that is, to listen for NTP multicasts for time adjustments), enter:

   ```
   # cp /etc/inet/ntp.client /etc/inet/ntp.conf
   # svcadm enable svc:/network/ntp:default
   # svcadm restart svc:/network/ntp:default
   ```

   For Windows 2003, NTP system behavior is controlled with the Group Policy MMC Snap-In. To make changes to the Windows Time Server, using the Group Policy Snap-In, and select **Computer Configuration**, click **Administrative Templates**, click **System**, and then click **Windows Time Service.** Here you can configure and enable time services for your Windows 2003 server(s.) Note that the Windows NTP service doesn't support multicasts.

   DNS – Active Directory is integrated with DNS, and as such, needs to fully configured and reliable. Client computers must be registed in both forward and reverse lookup zones. In lab environments, it may be easiest to allow the Active Directory setup wizard to install a DNS server. AD uses the SRV resource record type to locate Active Directory domain servers. In addition, the DNS server service should support dynamic DNS updates – if it doesn't, SRV records must be maintained manually by the administrator, which isn't practical. To ensure that DNS is able to return a domain controller address, enter the following command:

   ```
   # nslookup –query=any _ldap._tcp.domainname
   ```

   where *domainname* is the DNS domain name we're setting up, e.g. "example.com". This should return a SRV resource record like:

   _ldap._tcp.ttagov.com service = 0 100 389 win2k3.example.com.

   the information returned includes:

priority	'0' in the above	Clients attempt to connect to servers with the lowest priority
weight	'100' in the above	A load-balancing mechanism when more than one server has the same priority
port number	'389' in the above –	The port number where the server is listening to this service
hostname	'win2k3.example.com' in the above	The FQDN of the server this record pertains to

These values can be found/adjusted in the DNS Management snap-in.

There are other service record types as well, some other useful ones include:

nslookup –query=any _gc._tcp.*domainname*

locates a Global Catalog server for this domain.

nslookup –query=any _ldap._tcp._*SiteName*._sites._dc._msdcs.*domainname*

locates a domain controller in the site "SiteName" in the DNS domain "domainname".

2. Install NIS Server

This adds Unix attributes to the Active Directory schema

On an AD Catalog Server:
Select "Control Panel → Add/Remove Programs"
Click "Add/Remove Windows Components"
Select "Active Directory Services" → "Details" → "Identity Management for Unix" → "Server for NIS"
Ensure that "Administration Components" is also selected, then click "Ok", "Ok", then "Next" to install.

3. Install Windows Support Tools

This step installs the "ktpass.exe" utility, (used later in the procedure.)

Insert the Windows Server CD into your CD-ROM drive.
Click **No** if you are prompted to reinstall Windows.
When the Welcome screen appears, click **Perform additional tasks**, and then click **Browse this CD**.
Go to the \Support\Tools folder.

Double-click **suptools.msi**.
Follow the instructions that appear on your screen.

4. Create a service account in Active Directory

This account will be used to bind to Active Directory. This account need not have any special rights, and (if applicable) can be the same service account used by SGD to bind to Active Directory. Make sure the password never expires.

5. Create a AD user account for each Solaris "client" server

Use the Active Directory Users and Computers tool to create these accounts. Since Solaris will use the host service principal, a name like "host-solarissrvr" would be good.

For each account was created, run the ktpass.exe command to generate a unique keytab for each account (host). The command will look something like this (this example is for a hostname of "solx86.example.com", in the realm "TTAGOV.COM", with an output filename of "solx86.keytab):

```
\Program Files\Support Tools\ktpass -princ
host/solx86.example.com@EXAMPLE.COM -mapuser host-solx86 -crypto
DES-CBC-MD5 +DesOnly -pass password -ptype KRB5_NT_PRINCIPAL -out
solx86.keytab
```

Be sure to specify a unique output filename (so that you don't overwrite files; each server/account will needs its own unique file). I suggest using the server's name as the filename, i.e., something like "solarissrvr.keytab". Repeat the above for each Solaris server, and copy the resultant keytab files to the /etc/krb5 directory on each host as appropriate, naming the file as "/etc/krb5/krb5.keytab", owned by root, with mode 700 permissions. Note that if this file already exists, you will use the "ktutil" utility to merge these files.

6. Create a Global Security group for your Unix Users

Create a Group in Active Directory for your Unix users. This group will have the a scope of "Global" and a type of "Security". Once created, set the "Unix Attributes" of the created Group Name. When you create users, you'll make them members of this group.

7. Configure Kerberos

On each server, configure your Kerberos client as follows, replacing hostname(s), domains, and realm as appropriate. Be sure to use the proper case, and also be sure that comments ('#') begin in column 1, otherwise a rather obscure error message will result – trust me on this one.

```
# ident "@(#)krb5.conf  1.3     04/03/25 SMI"
# krb5.conf template
```

```
# In order to complete this configuration file
# you will need to replace the __<name>__ placeholders
# with appropriate values for your network.
#
[logging]
default = FILE:/var/log/kdc.log
kdc = FILE:/var/log/kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
   default_realm = EXAMPLE.COM
   dns_lookup_realm = false
   dns_lookup_kdc = false
   ticket_lifetime = 24h
   forwardable = yes

[realms]
EXAMPLE.COM = {
   kdc = win2k3.example.com
   default_domain = example.com
}

[domain_realm]
   .example.com = EXAMPLE.COM
   example.com = EXAMPLE.COM


[appdefaults]
   pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
}
```

Test your configuration with the following:

```
# kinit administrator
```

Enter the administrator password. If no error results, enter the "klist" command to confirm you have a valid ticket.

8. Configure the Solaris LDAP client

With a text editor, create the following shell script, replacing the “proxyDN” and “proxyPassword” values with the service account values you created previously. Be sure to use consistent naming within your “proxyDN” info – not the UPN or sAMAccountName. Also, change the “defaultServerList” to point to your domain controller. And, of course, change your domain to the proper values.

```
ldapclient manual  \
-a credentialLevel=proxy \
-a authenticationMethod=simple \
-a proxyDN=cn=Service Account Name,cn=Users,dc=example,dc=com \
-a proxyPassword= mypass \
-a defaultSearchBase=dc=example,dc=com \
-a domainName=example.com \
-a "defaultServerList=172.16.0.202" \
-a attributeMap=group:userpassword=userPassword \
-a attributeMap=group:memberuid=memberUid \
-a attributeMap=group:gidnumber=gidNumber \
-a attributeMap=passwd:gecos=cn \
-a attributeMap=passwd:gidnumber=gidNumber \
-a attributeMap=passwd:uidnumber=uidNumber \
-a attributeMap=passwd:homedirectory=unixHomeDirectory \
-a attributeMap=passwd:loginshell=loginShell \
-a attributeMap=shadow:shadowflag=shadowFlag \
-a attributeMap=shadow:userpassword=userPassword \
-a objectClassMap=group:posixGroup=group \
-a objectClassMap=passwd:posixAccount=user \
-a objectClassMap=shadow:shadowAccount=user \
-a serviceSearchDescriptor=passwd:dc=example,dc=com?sub \
-a serviceSearchDescriptor=group:dc=example,dc=com?sub
```

After this completes, you’ll have to edit /etc/nsswitch.conf to remove all references to “ldap” except for “user” and “group” entries. You may find it simpler to just copy “/etc/nsswitch.conf.bak” to “/etc/nsswitch” and manually add the “ldap” entry yourself.

Restart the LDAP client service:

```
# svcadm restart svc:/network/ldap/client:default
```

Test the configuration with a command like (substituting domain/service account info as appropriate):

```
ldapsearch -v -h 172.16.0.202 -b "dc=example,dc=com" \
-D "cn=SGD Service Account,cn=Users,dc=example,dc=com" -w - "cn=administrator"
cn
```

Specify the password for the service account at the "Enter bind password:" prompt. If successful, you'll see output like:

ldapsearch: started Mon Mar 26 14:44:41 2007

ldap_init(172.16.0.202, 389)
filter pattern: cn=administrator
returning: cn
filter is: (cn=administrator)
version: 1
dn: CN=Administrator,CN=Users,DC=example,DC=com
cn: Administrator
1 matches

9. Add Entries to PAM Configuration

Next we need configure the PAM configuration file. First backup your pam.conf file:

```
# cp /etc/pam.conf /etc/pam.conf.bak
```

Next edit /etc/pam.conf as follows:

```
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
other   auth requisite       pam_authtok_get.so.1
other   auth required        pam_dhkeys.so.1
other   auth sufficient      pam_krb5.so.1
other   auth required        pam_unix_cred.so.1
other   auth required        pam_unix_auth.so.1


#
# Default definition for Account management
#
other   account requisite       pam_roles.so.1
other   account sufficient      pam_unix_account.so.1
other   account required        pam_ldap.so.1
#
```

10. Reboot Server

After rebooting, you should be able to login using user credentials stored in the Active Directory directory service. At this juncture, common failures include:

- Missing Home Directory

- Invalid Shell
- Failed to set Unix attributes on the AD Account