An Oracle Technical White Paper

November 2014

How to Use Microsoft Active Directory as an LDAP Source with the Oracle ZFS Storage Appliance
Table of Contents

Introduction ..................................................................................................................3
Active Directory LDAP Services ..................................................................................4
Configuring the Oracle ZFS Storage Appliance for AD Access ...............................6
Configuring the Oracle ZFS Storage Appliance for LDAP Access ..........................9
  Verifying Expected Operation ................................................................................11
Complex AD Directory Structures ............................................................................16
Conclusion ..................................................................................................................20
Introduction

The Oracle ZFS Storage Appliance integrates advanced hardware and software architectures to offer a facile, multiprotocol storage system capable of running the most demanding workload. This workload includes a variety of simultaneously operating applications and advanced data services. First class performance characteristics are illustrated by the results of industry standard benchmarks like SPC-1, SPC-2 and SPECsfs.

As a unified storage platform, the Oracle ZFS Storage Appliance can be configured to operate in multiple environments concurrently. In order to fully integrate into these environments, it is necessary to subscribe to the naming systems appropriate to the particular environment.

For Microsoft Windows, this is usually Active Directory and for UNIX environments, LDAP is one of the most commonly deployed directory systems. In some circumstances, it is desirable to provide services to both Microsoft Windows and UNIX environments where Active Directory is the only deployment directory service.

This document describes how to use the Active Directory service to provide LDAP services that allow Identity Mapping to bridge the different environments.

For the purposes of this document, the Active Directory domain is called the fictional ‘example.org.’
Active Directory LDAP Services

Microsoft Active Directory is the industry-standard directory service for Microsoft Windows environments. Active Directory (AD) is a well-integrated combination of Kerberos for authentication, Lightweight Directory Access Protocol (LDAP) for authorization, and directory services and Domain Name System (DNS) for host name resolution and service location. AD is used to store information about users, groups, shares and many other types of shared objects.

The Oracle ZFS Storage Appliance integrates directly with AD to provide the consistent security and ownership details across Microsoft Windows environments. Out of the box, however, AD does not have a way to represent UNIX environment identities which are defined by user IDs (UIDs) and group IDs (GIDs) represented by positive integers. Other details such as home directories, group membership, default shell and encrypted passwords are also missing by default from AD. Some details (such as the GECOS field) can be borrowed from standard AD attributes.

By design, AD is extensible by schema modification. This means that it is possible to add attributes to objects where applications require, such as adding UIDs and GIDs to allow the Windows AD domain controller to manage UNIX environment directory accesses.

Windows Server versions up to and including Windows Server 2012 R2 have the facility to provide NIS directory access to UNIX hosts by installing Microsoft Identity Mapping for UNIX.

The standard for UNIX LDAP directory access is encapsulated in RFC 2307; installing Microsoft Identity Mapping for UNIX modifies the AD schema to comply with RFC 2307. There are a number of commercial packages that also provide the compliance with RFC 2307.

Active Directory’s LDAP service is not normally evident other than through the standard AD tools such as ADSI Edit, which allows access to the raw LDAP directory. It is through tools like these that the directory structure can be verified when the AD configuration varies from a simple out-of-the-box setup. An example of a newly created Active Domain is shown in Figure 1 in the “Active Directory Users and Computers” application and also in the more structured “ADSI Edit” in Figure 2.

![Figure 1. Active Directory Users and Computers view of Newly Created Domain](image-url)
The ADSI Edit view exposes more of the LDAP structure to the AD schema.

Other LDAP browsers are available, both commercially and as freeware, but since ADSI Edit is installed by default in Windows Server AD Server deployments, it will be used in this example.

In order to enable the UNIX attributes within Active Directory Users and Computers, you must create a blank NIS configuration, by importing empty text files in the IDMU NIS migration wizard. A tutorial covering use of Microsoft IDMU with the Oracle ZFS Storage Appliance is available at:

Configuring the Oracle ZFS Storage Appliance for AD Access

In order to create the appropriate authentication objects and permissions, the Oracle ZFS Storage Appliance should be configured to access AD. As mentioned previously, the DNS service plays a large role in AD and the Oracle ZFS Storage Appliance must therefore be configured to use the Windows DNS Service.

Clock synchronization is also important for correct AD operation. Clock skew between the Oracle ZFS Storage Appliance and the AD Domain Controllers must be less than 15 minutes for a successful AD join. It is highly recommended to install a Network Time Protocol (NTP) service and to configure the Oracle ZFS Storage Appliance and the AD domain controllers to become clients of the service.

1. Using the browser user interface (BUI) of the Oracle ZFS Storage Appliance, ensure that the DNS configuration refers to the same DNS server as the Active Directory servers. As shown in the following figure, access the DNS Configuration screen by selecting Configuration > Services > DNS.

![Figure 3. Verifying DNS configuration](image)

2. Ensure the clocks on the Oracle ZFS Storage Appliance and Windows AD Domain controllers are in synchronization.

In the BUI, select Configuration > Services > NTP. Enter the appropriate values for NTP Server(s) address details.

![Figure 4. Verifying clock synchronization](image)
3. From the Configuration > Services > Active Directory page, click the Join Domain button as shown in the following figure.

![Active Directory service page](image)

Figure 5. Active Directory service page

4. Enter the details of a Domain Administrator with the authority to allow the Oracle ZFS Storage Appliance to join the AD, as seen in figure 6. Click APPLY to continue.

![Join Domain](image)

Figure 6. Entering the AD administrator details

5. Upon a successful join operation, acknowledgement similar to the following is displayed.

![Successful AD Join](image)

Figure 7. Successful AD Join

If a message indicating ‘Access is denied’ is displayed, and the AD Domain Administrator username and password are correct, it may be necessary to change the LAN Manager compatibility level to Level 2.

Do so by selecting Configuration > Services > SMB and selecting the level from the drop-down towards the bottom of Properties as shown in Figure 8. Then you can return to step 3 to repeat the join domain process.
Figure 8. Configuring the LAN Manager compatibility level

The Oracle ZFS Storage Appliance is now able to resolve Windows environment users and groups from Active Directory. In addition, the necessary permissions are now in place for the Oracle ZFS Storage Appliance to access AD through the LDAP interface.
How to Use Microsoft Active Directory as an LDAP Source with the Oracle ZFS Storage Appliance

Configuring the Oracle ZFS Storage Appliance for LDAP Access

Given the example AD domain example.com, it is represented in the LDAP interface as the Distinguished Name (DN) DC=example,DC=org. This DN will be used as the base for searches.

In order to configure the Oracle ZFS Storage Appliance LDAP client, navigate to Configuration > Services > LDAP. Enter the ‘Base search DN,’ following the format just described (in this example, DC=example,DC=org).

Enable Subtree (or recursive) searches to allow the LDAP client to descend into the appropriate tree structures built up in AD.

It is possible to create a proxy user in AD to allow the Oracle ZFS Storage Appliance to access LDAP. However, as the previous section configured AD to allow access to LDAP, the bind credential level can be set to ‘Self,’ which will avoid the need to store the proxy user DN and password.

Next, the AD domain controllers should be added as Servers. In the following example, two AD domain controllers are nominated: w2k8-ad and w2k8-ad2.

The next step is to define the Schema definition.

By design, Windows users and groups share the same namespace – thus you cannot have a user and a group with the same name. By default these objects live in the ‘Users’ container which is represented in AD as CN=Users, <BASE-DN>. In the example, the container is CN=Users,DC=example,DC=org. This DN is then used as the User Search Descriptor in the Schema Definition.

UNIX environments, however, have separate namespaces for users and groups definitions, which allows groups to be created with the same name as users. To accommodate this difference, the group search descriptor must be set to the same as the user search descriptor.
The attributes `uidNumber` and `gidNumber` are defined in AD once IDMU is installed on the domain controllers and no mapping needs to occur. Other attributes are provided with slightly different names from the naming required by UNIX environments.

The following is a suggested list of the mappings:

<table>
<thead>
<tr>
<th>USER ATTRIBUTE</th>
<th>AD EQUIVALENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gecos</code></td>
<td><code>CN</code></td>
</tr>
<tr>
<td><code>homeDirectory</code></td>
<td><code>unixHomeDirectory</code></td>
</tr>
<tr>
<td><code>userPassword</code></td>
<td><code>unixUserPassword</code></td>
</tr>
<tr>
<td><code>uid</code></td>
<td><code>sAMAccountName</code></td>
</tr>
</tbody>
</table>

In order to allow these mappings to take place, it is necessary to map the UNIX object class `posixAccount` to the AD object class `User`. Similarly for group access, the UNIX object class `posixGroup` should be mapped to AD object class `group` with the search descriptor set to the same as that for users (due to the shared namespace AD employs.)

To implement these changes with the Oracle ZFS Storage Appliance BUI, navigate to **Configuration > Services > LDAP** (as shown in Figure 9) and select **Edit**… to the right of the 'Schema definition' line. This will bring up the 'Edit LDAP Schema Definition' dialog window as shown in Figure 10. Complete the following:

1. Enter the appropriate Search descriptor. The example shows `CN=Users,DC=example,DC=org`.
   
   Add the attribute mappings for `gecos`, `homeDirectory`, `userPassword` and `uid` either by using the suggested mappings or from those determined from your own AD schema. Click the plus icon to add any additional attribute mappings.

2. Enter the Object class mappings for `posixAccount` and `shadowAccount`. The latter must be different from the `posixAccount` as schema maps must be unique. Ultimately, mapping to person has the same effect.

3. Select the Group tab from the top left of the dialog box and enter the same search descriptor as in step 1.
Figure 11. Edit LDAP Schema Definition – Groups

4. Enter the Object class mapping for `posixGroup` and click **Save**.

Verifying Expected Operation

Once these schema definition changes have been made, check that the mapping occurs as expected in order to ensure that consistent ownership and permissions are maintained between Windows and UNIX environments.

**Note:** In order to test the correct mapping, there must be at least one user defined in AD with UNIX attributes configured. Similarly, there must be a security group in AD with its UNIX attributes configured prior to defining the user’s UNIX attributes.

For example, a group defined in AD as `unixusers` has been allocated the gid 10000. A user defined in AD as `andrew` has been allocated the uid 70592, with the primary group defined as `unixusers`.

Figure 12. UNIX Attributes for ‘andrew’

Another user has been defined in AD with the name `sharon`, and she does not have defined UNIX attributes.
From the Oracle ZFS Storage Appliance BUI, navigate to **Configuration > Services > Identity Mapping**. Ensure that the Mapping mode is set to **IDMU**. If necessary, change and **Apply** the change.

Select **Mappings**. From the Mappings screen, enter the Identity **andrew** and leave the Windows Domain at the default. Click the **SHOW** button.

As can be seen in Figure 14, user **andrew** does indeed have the correct user ID (70592.) The Group Properties can be ignored here because there is no group called **andrew** and thus an ephemeral ID has been generated but will never be used.

By selecting the Unix Platform radio box, the Windows ID can now be shown and, correctly, no group is displayed in this case.
Figure 15. Verifying mapping details for andrew - 2

Similar results display when selecting the Group Type radio box and entering unixusers.

Searching for sharon’s details produces no Windows details because, even though an ephemeral UID is automatically generated, there is no way to permanently map the Windows user ID to the ephemeral UNIX user ID (Figures 16 and 17).

Figure 16. Ephemeral IDs created for sharon
In order to provide a mapping between the UNIX and Windows environments for user Sharon, it is necessary to enter the UNIX attributes on the Active Directory Users and Computers wizard. In this example, her UNIX UID is 70593 and she is assigned to the group unixusers.

Once the changes have been applied, rerunning the mapping test shows the expected results as shown in Figure 19 and Figure 20.
Figure 19. Newly applied UID for sharon

Figure 20. Newly applied Windows ID for sharon
Complex AD Directory Structures

Where large and complex AD schemas have been deployed, Windows AD administrators may impose a hierarchy of users broken down by geography and/or function. It may be desirable then to limit the searches in certain circumstances to a single subtree or subtrees to allow optimum ID resolution speed by reducing the search space.

To illustrate this, the fictional example.org has over 20,000 active AD users defined spread across multiple countries. In order to maintain a manageable hierarchy, the AD administrators have implemented a geographically arranged tree. A subset of this tree concentrating on just the users is shown in Figure 21.

![Example.ORG hierarchy](image)

In order to keep the highest possible lookup performance, the storage administrators have decided that they will restrict searches to just the geographic areas and the administrative user subtrees.

Users exist within the country Organizational Units (OU) and these countries are contained in the geographical OU. A separate AdminUsers OU has also been created to allow group policies to be applied to these users without affecting the geographical users or vice versa.

As there is no common node other than the root of the domain and it would not be advisable to search the entire tree for users for performance reasons, the Oracle ZFS Storage Appliance offers an alternative approach in which the multiple subtrees can be nominated for searches by separating the DNs with a semicolon.

In the “simple” example, the user search descriptor was CN=Users,DC=example,DC=org. Using the additional search feature, the user search descriptor for the more complex configuration becomes:

```bash
OU=AdminUsers,DC=example,DC=org; OU=Americas,DC=example,DC=org; \ OU=APAC,DC=example,DC=org; OU=Europe,DC=example,DC=org
```

The Oracle ZFS Storage Appliance will then search each of these subtrees in turn to attempt to resolve any references. Thus, it may make sense to alter the order for the OUs, depending on where the Oracle ZFS Storage Appliances are sited, as well as which user base makes most use of the services offered by the Oracle ZFS Storage Appliance.
Additionally, the Group search descriptor should also be changed if groups are arranged in a similar manner – that is, under each geographical OU, or if there is a separate OU created to hold the security groups in AD, which might be the case where the functional unit assignment is defined in a global sense rather than geographical.

In the example.org domain, there are three users in the US OU and five in the UK OU, as shown in Figures 22 and 23 respectively.

![Figure 22. US OU Users](image1.png)

![Figure 23. UK OU Users](image2.png)

The ADSI Edit view of this configuration is shown in Figure 24.
With the User Search Descriptor set appropriately to capture the geographical units as well as the administrative users, IDMU can resolve and map from any of the nominated subtrees.

Some examples of this are shown in the following figures.

**Figure 24. ADSI Edit view**

**Figure 25. Resolving from AdminUsers**

**Figure 26. Resolving from AdminUsers**
Figure 27. Resolving from UK

Figure 28. Resolving from UK
Conclusion

The Oracle ZFS Storage Appliance provides a platform on which to bridge the gap between Windows and UNIX environments in a secure and consistent manner. Access authorization granted in one environment is mirrored in the other where the appropriate mappings are available.

By providing the flexibility to tailor the search descriptors, both simple and highly complex Active Directory schemas can be handled in a simple and consistent manner.

By providing the LDAP interfaces, the Identity Mapping feature brings together two disparate environments allowing the sharing of data and reduction in the number of storage islands where spare storage cannot be used in any other environment.