LdapServerIssues

LDAP Server Issues

The following are issues, encountered by some deployers, related to specific LDAP server products used in conjunction with username/password authentication or the attribute resolvers LDAP data connector.

Microsoft Active Directory

Port

Standard LDAP

If all users reside under the same single-depth object (e.g., `CN=Users,DC=example,DC=edu`), the standard ports can likely be used:

- **389** for plain-old LDAP or LDAP with StartTLS. Note, StartTLS is only available on Windows Server 2003 and later.
- **636** for LDAPS

Searches using the above connection information may encounter and need to handle referrals (see **Referrals** below).

Global Catalog

If users are spread across multiple object (e.g., `CN=Staff,DC=example,DC=edu` and `CN=Faculty,DC=example,DC=edu`) or if the standard connection method (above) doesn't work, the global catalog ports can be used:

- **3268** for plain-old LDAP or LDAP with StartTLS. Note, StartTLS is only available on Windows Server 2003 and later.
- **3269** for LDAPS

As a general note, the global catalog supports searches across the entire forest. Attributes that should be accessible to the Shibboleth IdP will have to be specified as part of the Partial Attribute Set (PAS) in Active Directory.

Bind DN

Active Directory authenticates users against its internal Kerberos realm. Therefore the principal name used for the `bindDn` configuration option should be a Kerberos principal name, `user@domain`, not a DN, `cn=user,ou=Users,dc=example,dc=org`.

Referrals

When performing a standard LDAP search on port 389/636, under some circumstances Active Directory will return LDAP referrals as a part of the LDAP result set. For example, this is known to occur when when using a domain DN as the LDAP search base (e.g. `dc=example, dc=org`) as opposed to a lower level container (e.g. `cn=Users,dc=example,dc=org`). These referrals must be followed for successful completion of the query by the connector. This entails adding a configuration parameter to the LDAP data connector configuration:

```
<LDAPProperty name="java.naming.referral" value="follow"/>
```

> The default value of `java.naming.referral` is "ignore". This causes the LDAP client to invoke the non-critical *Manage Referral* control (RFC 3296) when performing the LDAP search. This causes referrals to be returned as ordinary LDAP result entries rather than referral entries. As of this writing, Microsoft Active Directory does not support the *Manage Referral* control, and so always returns referral entries if they are applicable. This results in an **javax.naming.PartialResultException: Unprocessed Continuation Reference(s)** error if the client is not configured to follow referrals.

Note that the referrals themselves often contain LDAP server hostnames which differ from the original Active Directory LDAP server (domain controller) hostname. These hostnames must be resolveable via DNS in order for successful resolution. The referral service port must also be reachable via the network through firewalls, etc. If these conditions don't hold, a connection error will result, possibly reflected by either a **javax.naming. CommunicationException** or a **java.net.UnknownHostException** error message in the IdP log. In this case there are 2 options:

1. Determine why the referral can not be followed and address. It is often useful in this regard to use a CLI tool such as `ldapsearch` to perform the same LDAP query as the data connector, with the appropriate verbose output option to display the actual referral(s) that are being returned. In test environments or other environments where Microsoft DNS is not being used, it often turns out that the LDAP server hostname returned is not available to the IdP via DNS.
2. Reconfigure the data connector to instead use the AD Global Catalog (GC) as the source of attributes. This may be accomplished by using port 3268/3269 for the LDAP query rather than 389/636. This avoids the referral problem by ensuring that no referrals will be returned. However, this carries with it a couple of issues:
    - Not all AD domain controllers hold a copy of the GC.
    - The data available in the GC is not the full set of user attribute data, but rather only the partial attribute set that AD is configured to replicate to the GC. Current and future user attribute needs should be evaluated against the presence of the needed data in AD, and the willingness to configure the GC with additional attributes as needed. This is especially true if uncommon attributes or custom LDAP schema are to be used.

objectSid and objectGUID Attributes

Some deployments attempt to use the `objectSid` and `objectGUID` attributes and either release them directly or use them as a dependency in other attributes, such as persistent identifiers. Microsoft's documentation on these attributes is contradictory as to the type of the values of these attributes. Part of the documentation claims the values are binary (e.g. byte arrays) while another part claims the values are strings. If you are using these attribute **and** your Active Directory server treats these values a binary content you must configure the appropriate LDAP property to indicate that the values of these attributes are, in fact, binary data.

Failure to set this, in the case where these are binary values, is likely to cause the same 'unique' value to be returned for multiple accounts.

StartTLS

It is difficult to get definitive statements about StartTLSsupport in Windows Server operating systems

- Windows2000 is known to **not** support StartTLS
- Windows Server 2003 has been tested using IdP Release 2.3.6 and supports StartTLS
- Windows Server 2008 is assumed to support StartTLS

Sun Directory Server 6

This summary by no means covers all of the potential issues with using Sun DirectoryServer. This page describes the issues which were encountered during the upgrade from DirectoryServer 5.2 to DirectoryServer 6.3. Several issues caused service outages and had to be fixed before Shibboleth could continuously access the directory data without encountering occasional pauses.

StartTLS

DirectoryServer 6.3 has a bug which causes a 10x increase in the amount of time required to instantiate the StartTLS portion of a port 389 LDAP connection. Sun has a hotfix available for this issue (link to be added...).

Stale Connections

There is another bug, for which there is no hotfix as of this writing, which causes the LDAP server to "freak out" when it encounters a large number of connections suddenly made stale. This can happen for instance when a loadbalancer reaches a timeout and cuts off a large number of connections from various sources.

Shibboleth IdP 2.1 creates a new connection for every authentication request and leaves it open until garbage collection. This is soon to be fixed in an upgraded LDAP library, but prior to the fix, the problem is definitely noticeable. This, perhaps combined with other misbehaved applications, can result in large numbers of connections timed out at the same time. When this happens, the LDAP server takes a break to clean out all of these broken connections.   Delays of 8 solid minutes of outage have been observed where hundreds of connections were being cleaned out.

The solution for this issue, at least prior to the fix from Sun, is to enable the idle timeout in the LDAP server. This idle timeout should be shorter than other timeouts on the connection path such as those configured in a loadbalancer. When the connections are closed by the LDAP server, large numbers of connections can be timed out simultaneously within a second.

This bug is described in the sun forums here: http://forums.sun.com/thread.jspa?threadID=5322201

Sun DirectoryServer 5.2:

These changes were applied at USC and resulted in a 4x improvement in the use of DirectoryServer 5.2:

- Added "nsslapd-searchtune=57" to dse.ldif
- Applied a hotfix for bug 6508188
- Added "nsslapd-dncachesize: 350000" to dse.ldif