How to connect Isilon OneFS 7.x to LDAP and Active Directory

Contents

Introduction

Isilon OneFS supports multiprotocol data access, including SMB and NFS. Multiprotocol data access often coincides with networks that use several directory services. In the most common scenario, OneFS is connected to two directory services: Active Directory for Windows users connecting over SMB and LDAP for Unix users connecting over NFS.

Active Directory uses security identifiers (SIDs) to identify domain users, while LDAP uses user identifiers (UIDs) and group identifiers (GIDs) to identify domain users. For this reason, OneFS must map Active Directory SIDs to a corresponding LDAP UIDs and GIDs.

When a user with accounts in multiple directory services logs in to an Isilon cluster, the OneFS user mapper combines the user's identities from all directory services into an access token. OneFS uses this access token to identify the user and control access to folders and files.

Objective

This document explains how to connect OneFS to LDAP and Active Directory in combination, and how to configure OneFS with the two directory services for optimal system performance. To perform the procedures in this document, you should be familiar with the OneFS and UNIX commands, and you should be comfortable running commands from the OneFS command-line interface.

Step 1: Connect OneFS to LDAP

For networks that use LDAP and Active Directory in combination, Isilon recommends that you configure LDAP first; this is to prevent issues with UID coherency.

> **Note:** If your cluster was connected to Active Directory before it was connected to LDAP, you can contact EMC Isilon Technical Support for help cleaning up user mappings; this will prevent possible issues with UID coherency from occurring. Then you can continue to "Step 3: Configure OneFS settings for LDAP and Active Directory."

To connect OneFS to an LDAP directory service, complete the following steps. At a minimum, `ldap-server-uri` and `ldap-base-dn` need to be defined.

1. Open an SSH connection on any node in the cluster and log in using the "root" account or an account with the ISI_PRIV_AUTH privilege.

2. Run the following command to configure an LDAP server, where `<ldap server>` is the name of the LDAP server, `<server URI>` is the uniform resource identifier (URI) for the LDAP server including "ldap://" or "ldaps://", `<domain name>` is the LDAP domain name, and `<domain extension>` is the domain extension:

   ```
   isi auth ldap create <ldap server> --server-uris=<server
   URI> --base-dn=dc=<domain name>,dc=<domain extension>
   ```

For example, run the following command to connect to the LDAP server ladpusers.example.com:

```
isi auth ldap create idsrv --server-uris=
ldap://idsrv.example.com --base-dn=dc=example,dc=com
```

The OneFS LDAP provider uses the `base-dn` when retrieving user and group information; therefore, the `ldap-base-dn` must be configured as a DN that contains both sets of information. For instance, on idsrv.example.com, user information is contained in `OU=people,dc=example,dc=com`. Group information is contained in `OU=groups,dc=example,dc=com`. By setting `ldap-base-dn` to `dc=example,dc=com`, the user and group information is encapsulated. If reading user and group information from your LDAP server requires an authorized user, the `ldap-bind-dn` and `ldap-bind-pw` should be set accordingly.

Verification

You can verify the properties of an LDAP user by running the following command, where <user> is the user account you want to look up:

```
isi auth users view <user>
```

For example:

```
isi auth users view janedoe
```

Output similar to the following appears:

```
Name: janedoe
DN: uid=janedoe,ou=people,dc=example,dc=com
DNS Domain: -
Domain: LDAP_USERS
Provider: lsa-ldap-provider:idsrv
Sam Account Name: janedoe
UID: 10028
SID: S-1-22-1-10028
Enabled: Yes
Expired: No
Expiry: - Locked: No Email: -
GECOS: Jane Doe Generated GID: No Generated UID: No Generated UPN: -
Primary Group
  ID : GID:10000
  Name : example
Home Directory: /ifs/home/janedoe
Max Password Age: Never
Password Expired: No
Password Expiry: -
Password Last Set: -
Password Expires: Yes
Shell: /bin/sh
UPN: -
User Can Change Password: No
```

The UID and GID come from LDAP, and the SID is generated from the UID. To confirm that this output is from LDAP, make sure that the Domain is LDAP_USERS.

Step 2: Connect OneFS to Active Directory

To connect an Isilon cluster to Active Directory service and join an Active Directory domain, complete the following steps:

1. Open an SSH connection on any node in the cluster and log in using the "root" account, or an account with the ISI_PRIV_AUTH privilege.

2. Run the following command to create an Active Directory provider, where <domain> is the fully-qualified domain name and <user> is an Active Directory user name with permission to join machines to the given domain:

   ```
   isi auth ads create <domain> <user>
   ```

   For example:

   ```
   isi auth ads create EXAMPLE.COM Administrator
   ```

When Active Directory and LDAP are used together with default mapping rules, a user's UID is retrieved from LDAP; the user's GID is an automatically assigned value for the default group in Active Directory; and the user and primary group SIDs are from Active Directory. The user's supplemental groups are a combination of what is returned from Active Directory along with groups found in LDAP.

Verification

You can verify the properties of an Active Directory user by running the following command, where <domain> is the Active Directory domain and <user> is the user that you want to look up. Make sure to include the quotation marks.

```
isi auth users view "<domain>\\<user>"
```

For example:

```
isi auth users view "EXAMPLE\\janedoe"
```

Output similar to the following appears:

```
Name: EXAMPLE\janedoe
DN: CN=janedoe,CN=Users,DC=example,DC=com
DNS Domain: example.com
Domain: EXAMPLE
Provider: lsa-activedirectory-provider:EXAMPLE.COM
Sam Account Name: janedoe
UID: 10028
SID: S-1-5-21-1234567890-1234567890-1234567890-1234
Enabled: Yes
Expired: No
Expiry: -
Locked: No
Email: -
GECOS: janedoe
```

```
Generated GID: Yes
Generated UID: No
Generated UPN: No
Primary Group
  ID : GID:1000000
  Name : EXAMPLE\domain users
Home Directory: /ifs/home/EXAMPLE/janedoe
Max Password Age: -
Password Expired: No
Password Expiry: -
Password Last Set: 2012-05-09T15:54:29
Password Expires: No
Shell: /bin/zsh
UPN: janedoe@EXAMPLE.COM
User Can Change Password: Yes
```

It is worth noting that the home directory, gecos, and shell values are from Active Directory. This is expected when Active Directory and LDAP are used together with default mapping rules.

Step 3: Configure OneFS settings for LDAP and Active Directory

After connecting the Isilon cluster to LDAP and Active Directory, you should configure OneFS for optimal system performance. This section includes the OneFS commands and recommended settings to use in an LDAP and Active Directory multiprotocol environment. You can use these commands to set preferences for user authentication and file access options. For a description of these and other commands used in OneFS, see the OneFS 7.1 CLI Administration Guide.

isi auth ads modify

You can use the `isi auth ads modify` command to modify an Active Directory provider and ACL policy settings. For additional information about the parameters and options that are available for this command, run the `isi auth ads modify --help` command.

Syntax

```
isi auth ads modify <provider-name>

    [--allocate-gids {yes | no}]
    [--allocate-uids {yes | no}]
    [--lookup-domains <dns-domain>]
    [--lookup-groups {yes | no}]
    [--lookup-normalize-groups {yes | no}]
    [--lookup-normalize-users {yes | no}]
    [--users {yes | no}]
```

Table 1. Options and recommended settings to use with isi auth ads modify

Command Option	Description	Default	Recommended
`--allocate-gids`	Enables or disables automatic GID allocation for unmapped Active Directory groups. If this option is disabled (recommended), GIDs are not proactively assigned, but when a user's primary group does not include a GID, the system may allocate one.	Yes	No
`--allocate-uids`	Enables or disables automatic UID allocation for unmapped Active Directory users. If this option is disabled (recommended), UIDs are not proactively assigned, but when a user's identity does not include a UID, the system may allocate one.	Yes	No
`--lookup-domains`	Restricts user and group lookups to the specified domain. If this setting is not set (recommended), UID and GID lookups for Active Directory users will be done from the primary domain.	Undefined	Undefined
`--lookup-groups`	Specifies whether to look up Active Directory groups in other providers before allocating a GID.	Yes	Yes
`--lookup-normalize-groups`	Specifies whether to normalize Active Directory group names to lowercase before looking them up.	Yes	Yes
`--lookup-normalize-users`	Specifies whether to normalize Active Directory user names to lowercase before looking them up.	Yes	Yes
`--lookup-users`	Specifies whether to look up Active Directory users in other providers before allocating a UID.	Yes	Yes

isi auth settings global modify

You can use the `isi auth settings global modify` command to configure access management settings, such as whether to automatically allocate UIDs and GIDs in the ID mapper, and what the ranges should be for UIDs and GIDs. For additional information about the parameters and options that are available for this procedure, run the `isi auth ads modify --help` command.

Syntax

```
isi auth settings global modify

    [--gid-range-enabled {yes | no}]
    [--gid-range-min <integer>]
    [--gid-range-max <integer>]
    [--uid-range-enabled {yes | no}]
    [--uid-range-min <integer>]
    [--uid-range-max <integer>]
    [--on-disk-identity {native | unix | sid}]
```

Table 2. Options and recommended settings to use with isi auth settings global modify

Command Option	Description	Default	Recommended
`--gid-range-enabled`	Enables or disables the automatic allocation of GIDs in the ID mapper. Use `--gid-range-min` and `--gid-range-max` to modify the default range.	Yes	Yes
`--gid-range-min`	Specifies the lower limit of the GID range if `--gid-range-enabled` is set to `yes`.	1000000	1000000
`--gid-range-max`	Specifies the upper limit of the GID range if `--gid-range-enabled` is set to yes.	2000000	2000000
`--uid-range-enabled`	Enables or disables the automatic allocation of UIDs in the ID mapper. Use `--uid-range-min` and `--uid-range-max` to modify the default range.	Yes	Yes
`--uid-range-min`	Specifies the lower limit of the GID range if `--uid-range-enabled` is set to yes.	1000000	1000000
`--uid-range-max`	Specifies the upper limit of the GID range if `--gid-range-enabled` is set to `yes`.	2000000	2000000
`--on-disk-identity`	Controls the preferred identity to store on disk. If OneFS is unable to convert an identity to the preferred format, it is stored as is. This setting does not affect identities that are already stored on disk.	Native	Native

isi nfs exports modify

You can use the `isi nfs exports modify` command to modify an NFS export.

Syntax

```
isi nfs exports modify

    [--map-lookup-uid {yes | no}]
```

Table 3. Options and recommended settings to use with isi nfs exports create

Command Option	Description	Default	Recommended
`--map-lookup-uid`	If set to `yes` (recommended), incoming UNIX user identifiers (UIDs) will be mapped to a full user token including Windows SIDs. In this way, NFS file access behaves the same as SMB file access.	No	Yes

Step 4: Configure User Mapping Rules

When a user with accounts in Active Directory and LDAP logs in to an Isilon cluster, the user mapper service combines the user's identities and privileges from all the directory services into an access token. OneFS uses this access token to identify the user and control access to directories and files.

You can create mapping rules to modify access tokens. For example, you can create a rule to merge an Active Directory identity and an LDAP identity into a single token that works for access to files stored over both SMB and NFS.

For a detailed discussion of the user mapping service and mapping rules, see Identities, access tokens, and the Isilon OneFS user mapping service.

Default mappings

If no mapping rules are set up, a user authenticating with one directory service receives full access to the identify information in other directory services when the account names are the same. For example, an Active Directory user who authenticates as Desktop\jane receives identities for the corresponding jane LDAP user account.

When OneFS is connected to Active Directory and LDAP directory services, the default mapping provides a user with a UID from LDAP, a user SID from Active Directory, a group SID from the default group in Active Directory, and a GID of an LDAP group that matches the Active Directory group; otherwise, a GID is allocated automatically. The user's groups come from Active Directory and LDAP, with the LDAP groups added to the list. The user's home directory, gecos, and shell come from Active Directory.

Suggested mapping rules in a multiprotocol environment

The following mapping rules are suggested when using both Active Directory and LDAP directory services.

> **Note:** Rules 1 and 2 are mutually exclusive and should not be added together. Generally, Rule 1 is appropriate for most cluster configurations.

1. Primary user is from Active Directory, and primary group is from LDAP

```
isi zone zones modify --zone=System --add-user-mapping-rules="DOMAIN\* += *
[group]"
```

2. Authenticate users with Active Directory, but check file access with LDAP identity

```
isi zone zones modify --zone=System --add-user-mapping-rules="DOMAIN\* => *"
```

3. Prevent users who are not listed in both LDAP and Active Directory from connecting to the cluster

This scenario includes two rules.

```
isi zone zones modify System --add-user-mapping-rules="*\* += * [group]"

isi zone zones modify System --user-mapping-mapping=rules="<default_unix_user=this-
user-does-not-exist>"
```

The first rule replaces the primary group GID of an Active Directory user with the GID of the corresponding UNIX user in LDAP.

If a user does not exist in LDAP, OneFS looks up the default_unix_user to obtain the GID. Because the default_unix_user maps to a nonexistent user, the lookup fails, and the authentication also fails.