IBM Security Verify Access
Version 10.0.8
June 2024

User registry configuration topics

Contents

Tables

Chapter 1. Supported registries

Security Verify Access supports several user registries and their supported operating systems.

Refer to the Software Product Compatibility Report for a complete list of supported user registries. The user registry support is located in **related software** > **Identity Management**.

Chapter 2. User registry server installation and configuration

Set up a registry server for use with Security Verify Access so that you can establish a management domain.

Review the information in user registry considerations.

To install and configure a registry, do one of the following tasks:

- To install and configure IBM® Security Directory Server version 6.4, which is included with Security Verify Access, see IBM Security Directory Server v6.4 documentation.

  The instructions in the following topics are for Directory Server version 6.3.

  - “Installing Security Directory Server with a wizard” on page 7
  - “Installing Security Directory Server with a script (AIX, Linux, Solaris)” on page 10
  - “Installing Security Directory Server with a script (Windows)” on page 13
  - “Installing Security Directory Server with Launchpad (Windows)” on page 15

  You also can consult the IBM Security Directory Server documentation available on the web at:

  http://www.ibm.com/software/tivoli/products/directory-server
- To install a supported registry other than IBM Security Directory Server, use the registry product's documentation. For a list of supported registries, see Product requirements. Ensure that all necessary operating system patches are installed.
- To use an existing registry server with Security Verify Access, ensure that you upgraded the server to a version that is supported by this release of Security Verify Access. For other supported registries, consult the registry product's documentation. Follow the instructions to configure your registry for use with Security Verify Access.

User registry considerations

Security Verify Access supports several LDAP registries. Before you configure a registry, consider the naming limitations, how the LDAP registry works with Security Verify Access in your environment, and the requirements for specific registries.

Supported LDAP user registries

A comprehensive range of LDAP user registries are supported, however note that all registries can be configured as the primary registry for Security Verify Access. Where the primary registry is used to store traditional ISVA user data and requires a custom schema to be configured.

Evaluate the supported registry versions based on the product requirements maintained at the following website: http://www.ibm.com/software/reports/compatibility/clarity/index.html

Note: Where the supported registry has reached End of standard Support (See the corresponding vendors product lifecycle), support for compatibility/interoperability are offered on a best effort basis and are not tested as part of a routine release cycle. Additionally, security enhancements (including cipher suites or other product changes) may limit or prevent interoperability across some versions.

Name limitations for supported registries

Review this information before you configure a user registry for your environment.

- Avoid forward slashes (/) in names for users and groups when you define that name with distinguished names strings. Each user registry treats this character differently.

- Avoid leading and trailing blanks in user and group names. Each user registry treats blanks differently.

LDAP configuration information

Review this information before you configure an LDAP registry for your environment.

- Security Verify Access requires no configuration steps so that it supports LDAP's Password Policy. It does not assume the existence or non-existence of LDAP's Password Policy.
  - Security Verify Access enforces its own Password Policy first. Security Verify Access attempts to update password in LDAP only when the provided password passes Security Verify Access's own Password Policy check.
  - A Security Verify Access tries to accommodate LDAP' Password Policy with the return code that it receives from LDAP during a password-related update.
  - If Security Verify Access can map this return code without any ambiguity with the corresponding Security Verify Access error code, it does so and returns an error message.
- To take advantage of the multi-domain support in Security Verify Access, you must use an LDAP user registry.
- With an LDAP user registry, the capability to own global sign-on credentials must be explicitly granted to a user. After this capability is granted, you can remove it.
- Leading and trailing blanks in user names and group names are ignored in an LDAP user registry in a Security Verify Access secure domain. To ensure consistent processing regardless of the user registry, define user names and group names without leading or trailing blanks.
- Attempting to add a single duplicate user to a group does not produce an error in an LDAP user registry.
- The Security Verify Access authorization API provides a credentials attribute entitlements service. This service is used to retrieve user attributes from a user registry. When this service is used with an LDAP user registry, the retrieved attributes can be string data or binary data.

LDAP data format

The following LDAP data formats are available for user and group tracking information.

Minimal
: **Minimal** format uses fewer LDAP objects to maintain user and group tracking information. Using this format reduces the size of your user registry information because minimal user and group tracking information is stored.

Standard
: **Standard** format uses more LDAP objects to maintain user and group tracking. This format was also used in versions of IBM Tivoli® Access Manager for e-business before version 6.0.

If the user and group information in the LDAP registry is used by other Security Verify Access products, such as IBM Tivoli Access Manager for Operating Systems or the Federation Runtime, the same LDAP data format must be used for all products.

Sun Java System Directory Server look-through limit

When the directory server is installed, the default value for look-through limit is 5000. If the user registry contains more entries than the defined look-through limit, the directory server might return the following status that Security Verify Access treats as an error:

```
LDAP_ADMINLIMIT_EXCEEDED
```

You can modify this value from the Sun Java™ System Directory Server Console:

1. Select the **Configuration** tab.
2. Expand the **Data** entry.
3. Select **Database Settings**.

4. Select the **LDBM Plug-in Settings** tab.
5. In the **Look-through Limit** field, type one of the following responses:
   - The maximum number of entries that you want the server to check in response to the search, or type
   - `-1` to define no maximum limit.

   Note: If you bind the directory as the Directory Manager, the look-through limit is unlimited and overrides any settings that are specified in this field.

Microsoft Active Directory Lightweight Directory Services (AD LDS)

Review this information before you configure a Microsoft AD LDS registry for your environment.

- For IBM Security Access Manager version 9, there is no option for standard or minimal data model. Standard data model is only available for a migrated policy server. Because AD LDS requires a single naming attribute for creating LDAP objects, AD LDS requires the minimal data model. Regardless of which data model you choose, Security Verify Access always uses the minimal data model if you select AD LDS as the user registry.

  The common name (**cn**) attribute is a single-value attribute and can store only one value. The AD LDS registry requires the value of **cn** to be the same as the **cn** naming attribute in the distinguished name (**dn**) attribute. When you create a user or group in Security Verify Access, specify the same value for **cn** as the **cn** naming attribute in the **dn**. Security Verify Access ignores the value of the **cn** attribute if it is different from the value of the **cn** naming attribute in the **dn**. For example, you cannot use the following command to create a user because the value of the **cn** attribute, `fred`, is different from the **cn** naming attribute in the **dn**, `user1`:

  ```
  pdadmin user create user1 cn=user1,o=ibm,c=us fred smith password1
  ```

Maximum lengths for names by user registry

The maximum lengths of various names that are associated with Security Verify Access vary depending on the user registry in the environment.

Table 1 on page 5 shows the maximum lengths that are allowed for each user registry that is supported by Security Verify Access. Maintaining these maximum lengths ensures compatibility.

Table 1. Maximum lengths for names by user registry and the optimal length across user registries

Name	IBM Security Directory Server	IBM z/OS® Security Server	Novell eDirectory Server	Sun Java™ System Directory Server	Microsoft Active Directory Server	Active Directory Lightweight Directory Service (ADLDS)	Optimal length
First name (LDAP CN)	256	256	64	256	64	64	64
Middle name	128	128	128	128	64	64	64
Last name (surname)	128	128	128	128	64	64	64
Registry UID (LDAP DN)	1024	1024	1024	1024	2048	1024	255
Security Verify Access user identity	256	256	256	256	64	64	64

Table 1. Maximum lengths for names by user registry and the optimal length across user registries (continued)

Name	IBM Security Directory Server	IBM z/OS® Security Server	Novell eDirectory Server	Sun Java™ System Directory Server	Microsoft Active Directory Server	Active Directory Lightweight Directory Service (ADLDS)	Optimal length
User password	unlimited	unlimited	unlimited	unlimited	256	128	256
User description	1024	1024	1024	1024	1024	1024	1024
Group name	256	256	256	256	64	64	64
Group description	1024	1024	1024	1024	1024	1024	1024
Single sign-on resource name	240	240	240	240	60	240	60
Single sign-on resource description	1024	1024	1024	1024	1024	1024	1024
Single sign-on user ID	240	240	240	240	60	240	60
Single sign-on password	unlimited	unlimited	unlimited	unlimited	256	unlimited	256
Single sign-on group name	240	240	240	240	60	240	60
Single sign-on group description	1024	1024	1024	1024	1024	1024	1024
Action name	1	1	1	1	1	1	1
Action description, action type	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
Object name, object description	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
Object space name, object space description	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
ACL name, ACL descriptions	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
POP name, POP description	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited

Although the maximum length of an Active Directory distinguished name (registry UID) is 2048, the maximum length of each relative distinguished name (RDN) is 64.

If you configure Security Verify Access to use multiple Active Directory domains, the maximum length of the user identity and group name does not include the domain suffix. When you use multiple domains, the format of a user identity is *user_id*@*domain_suffix*. The maximum length of 64 applies only to the *user_id* portion. If you use an email address or other alternative format for the Security Verify Access user identity in the Active Directory, the maximum name length remains the same, but includes the suffix.

Although the lengths of some names can be of unlimited, excessive lengths can result in policy that is difficult to manage and might result in poor system performance. Choose maximum values that are logical for your environment.

Security Directory Server installation and configuration

Security Directory Server is provided with the Security Verify Access product. You can use a new installation or an existing installation of Security Directory Server in your environment.

Review the information in user registry considerations. Then, choose an installation method. If you use an existing registry server with Security Verify Access, ensure that you upgraded the server to a version that is supported by this release of Security Verify Access. For other supported registries, consult the registry product's documentation. Then, follow the instructions to configure your registry for use with Security Verify Access.

Installing Security Directory Server with a wizard

Install IBM Security Directory Server by using the installation wizard in the typical installation path. It uses default values and automatically installs all the required Security Directory Server components for Security Verify Access.

Before you begin

Note:

- The information in this topic is for Directory Server version 6.3. Use the information and links in the tech note for the configuration instructions for Security Directory Server 6.4 and Security Directory Suite 8.0.
- If Security Directory Server packages, such as client packages, are already installed at a level greater than 6.3.0.0, remove the packages before you run the installation wizard.

Complete the following tasks before you set up IBM Security Directory Server:

- Review the user registry considerations.
- Access the instructions for the `Typical installation path` method in the IBM Security Directory Server version 6.3 IBM Knowledge Center.
  1. Go to the IBM Security Directory Server version 6.3 IBM Knowledge Center.
  2. Search for `Typical installation path`.

About this task

This task completes installations of the following components:

- All components that are required by Security Directory Server.
- All the corequisite products that are required by Security Directory Server, if they are not already installed.
  - GSKit
  - DB2

- The embedded version of WebSphere Application Server. This software is required by the Web Administration tool, which is installed automatically as part of the `Typical installation path` method.

This task also completes the following configuration:

- Deploys the Web Administration tool.
- Creates a default directory server instance named **dsrdbm01**.
- Creates the operating system user ID named **dsrdbm01** that owns the instance.
- Creates an Administrator DN named **cn=root**.
- Creates a default suffix named **o=sample**.

Procedure

1. Log on to the system.

   AIX, Linux, or Solaris
   Log on as `root`.

   Windows
   Log on as an administrator.
2. Use the following steps to prepare and start the installation program:
   a) Access the DVD or extract the files from the archive file that you downloaded from Passport Advantage.
   b) For AIX, Linux, or Solaris systems: Install the Security Directory Server license files by running the `idsLicense` script in the *`image_path`*`/tdsV6.3FP/license` directory, where *image_path* is the path to the DVD image, or where you downloaded the archive file from Passport Advantage®.
   c) Change to the *`platform`*`/tdsV6.3/tds` directory.
3. Run the installation program.

   AIX, Linux, or Solaris
   Run `install_tds.bin`.

   Windows
   Double-click the **install_tds.exe** icon.
4. Complete the installation by using the `Typical installation path` instructions in the IBM Security Directory Server IBM Knowledge Center.

   For more information, see the Security Directory Server Knowledge Center.

   Note: Record any passwords that you set during the installation so that you can use them in subsequent installation steps.
5. Complete the following steps when the Security Directory Server Instance Administration tool opens.
   a. Verify that the default instance is listed in the configuration.

      Note: If you are using Red Hat Enterprise Linux 6, the default instance is not displayed in the tool. To verify that it is listed in the configuration, use the **`idsilist`** command. See the IBM Security Directory Server version 6.3 IBM Knowledge Center for details about the command. By default, this command is in `/opt/ibm/ldap/V6.3/sbin/`.
   b. Do not start the instance.
   c. Exit the tool.
6. Start the configuration process by using the command line. Create the suffix where Security Verify Access maintains its metadata with the **`idscfgsuf`** command.

   ```
   idscfgsuf -s "secAuthority=domain_name"
   ```

   The command is in the following locations by default:

AIX, Linux, or Solaris
`/opt/ibm/ldap/V6.3/sbin/idscfgsuf`

Windows
`c:\Program Files\IBM\LDAP\V6.3\sbin\idscfgsuf`

where *domain_name* is the management domain name.

The default suffix is Default; for example:

```
idscfgsuf -s "secAuthority=Default"
```

If you specify a location for the metadata that is not a stand-alone suffix, ensure that the location exists in the LDAP server.

This suffix is added to the `ibmslapd.conf` file for the default instance. If you have more than one instance, specify the instance name by using the *-I* option.

7. Optional: You can create more suffixes to maintain user and group definitions.

```
idscfgsuf -s "c=US"
```

8. Start the LDAP server.

AIX, Linux, or Solaris

```
ibmslapd&
```

Windows
From the **Services** window, start the following services:

```
IBM Security Directory Server Instance V6.3 - instance_name
```

9. Optional: For AIX, Linux, or Solaris systems only: Update the installation to the appropriate fix pack level.

Note: For Windows installations, the installation image includes the appropriate fix pack level.

a) Stop all Security Directory Server services.
b) Access the DVD or extract the files from the archive file that you downloaded from Passport Advantage.
c) Change to the appropriate directory for your operating system.

```
platform/tdsV6.3FP
```

d) See the readme file that is included with the fix pack for information and installation instructions.
e) Run the installation program.

```
./idsinstall –u -f
```

10. When the installation completes, verify the installed versions.
a) Open a command prompt.
b) Type:

```
idsversion
```

What to do next

If you are setting up SSL communication, see “Configuring IBM Tivoli Directory Server for SSL access” on page 17.

Installing Security Directory Server with a script (AIX, Linux, Solaris)

Use the script file to automate the installation of Security Directory Server.

Before you begin

The information in this topic is for Directory Server version 6.3. Use the information and links in the tech note for the configuration instructions for Security Directory Server 6.4 and Security Directory Suite 8.0.

About this task

Automated installations can complete unattended silent installations. This task uses the **idsNative Install** command.

Procedure

1. Log on to the system with root privileges.
2. Access the product DVD or extract the files from the archive file that you downloaded from Passport Advantage.
3. Extract the Security Directory Server archive file to a directory with adequate disk space.
   For example, `/tdsV6.3/`. If you use a DVD to install Security Directory Server, the files are in the `tdsV6.3` directory.
4. Locate the following script files and change the permissions so that you can write to the files:

   ```
   chmod +w image_path/tdsV6.3/responsefile.txt
   chmod +w image_path/scripts/ISAMConfigTDS.sh
   chmod +w image_path/scripts/ISAMGenSSLCert.sh
   chmod +w image_path/platform/tdsV6.3/idsConfigServerSSL.sh
   ```

5. Install the Security Directory Server license files by completing the following steps:
   a. Navigate to the *image_path*`/tdsV6.3FP/license` directory.
   b. Run the following script:

      ```
      idsLicense -q
      ```

      where the `-q` option installs the license files without displaying the license. If you use the `-q` option, you automatically accept the license without viewing it.
6. In the `tdsV6.3` directory, locate the installation program file and the response file:
   - `idsNativeInstall.sh`
   - `responseFile.txt`

   These files must be in the same directory.
7. Update the following entries in the `responseFile.txt` file.
   By default, the values of the variable are set to `false` and their corresponding path variables are not set.
   - To install DB2, set the *db2FeatureInstall* variable to `true`. Update the *db2InstallimagePath* variable with the absolute path where the DB2 installation files are located.

     For example:

     ```
     db2FeatureInstall=true
     db2InstallimagePath=/image_path/platform/tdsV6.3/db2
     ```

   - To install GSKit, set the *gskitFeatureInstall* variable to `true`. Update the *gskitInstallimagePath* with the absolute path to where the GSKit installation files are located. For example:

     ```
     gskitFeatureInstall=true
     gskitInstallimagePath=/image_path/platform/tdsV6.3/gskit
     ```

- To install embedded WebSphere Application Server (eWAS), set the *eWasFeatureInstall* variable to `true`. Update the *eWasInstallimagePath* with the absolute path to where the embedded WebSphere Application Server installation files are located. For example:

```
eWasFeatureInstall=true
eWasInstallimagePath=/image_path/platform/tdsV6.3/appsrv
```

- To install Security Directory Server, update the *tdsInstallimagePath* with the absolute path to where the Security Directory Server installation files are located. Update the *tdsFixPackInstallimagePath* variable with the absolute path to where the Security Directory Server fix pack installation files are located. For example:

```
tdsInstallimagePath=/image_path/platform/tdsV6.3/
tdsFixPackInstallimagePath=/image_path/platform/tdsV6.3FP
```

Note: If you want to install the full Security Directory Server , but there are already some Security Directory Server packages installed, such as the client packages, remove the images before you run this script.

8. Save the `responseFile.txt` file.
9. For Solaris systems only:
   a. Check that the `/export/home` directory exists. If the directory does not exist, create it.
   b. Ensure that the following kernel parameters in the `/etc/system` file are set appropriately for your system. The following values are suggested as starting values:

```
set msgsys:msginfo_msgmax = 65535
set msgsys:msginfo_msgmnb = 65535
set shmsys:shminfo_shmmax = 2134020096
```

   For more information, see the Solaris tuning documentation.
10. Open a command prompt and start the installation by typing **idsNativeInstall.sh**
11. Verify the installation by checking the installation log:
    `/var/idsldap/V6.3/idsNativeInstall_`*timestamp*`.log`
12. For AIX, Linux, or Solaris systems only: Update the installation to the appropriate fix pack level.

    Note: For Windows installations, the installation image includes the appropriate fix pack level.

    a) Stop all Security Directory Server services.
    b) Access the DVD or extract the files from the archive file that you downloaded from Passport Advantage.
    c) Change to the appropriate directory for your operating system.

```
platform/tdsV6.3FP
```

    d) See the readme file that is included with the fix pack for information and installation instructions.
    e) Run the installation program.

```
./idsinstall –u -f
```

13. Optional: If you want to use the Security Directory Server Web Administration Tool, deploy Security Directory Server into the embedded version of WebSphere Application Server:
    a) Open a command prompt.
    b) Run *ldaphome*`/idstools/deploy_IDSWebApp`.
    Replace *ldaphome* with the installation path.
14. Create the default instance and suffix:
    a) Open a command prompt.
    b) Change to the following directory: *image_path/platform*`/tdsV6.3/`
    c) Run the following command:

```
idsdefinst -p passworddn -w passworduser -e encryptseed
```

where:

passworddn
The administration DN password. For example, `cn=root` password.

passworduser
The database owner password. For example, the password for the user ID `dsrdbm01`.

encryptseed
The encryption seed value. This value is used to create is used to generate a set of Advanced Encryption Standard (AES) secret key values. The length must be 12 - 1016 characters.

15. Configure Security Directory Server for Security Verify Access:
    a) Locate the *image_path*`/scripts/ISAMConfigTDS.sh` file.
    b) Open the file in a text editor.
    c) Set the `adminPW` to the `cn=root` password.
    This password was created when the **idsdefinst** tool was run.
    d) Review the other settings in the file. If you used the default values during the installation of Security Directory Server, no further modification is required.
    e) Save and close the `ISAMConfigTDS.sh` file.
    f) Open a command prompt.
    g) Run *image_path*`/scripts/ISAMConfigTDS.sh`.
    Replace *image_path* with the path to the script files.
    h) Review output messages and verify that the script completed successfully.

    Note: If you used an improper database name, the script might exit with a return code of zero. Review all messages to ensure that the script completed successfully. The default database name is `dsrdbm01`. You do not need to change the default name if you used the defaults with the **idsdefinst** command.
16. Optional: If you are setting up Suite B and NIST compliance between your user registry and Security Verify Access components, see "Configuring IBM Tivoli Directory Server for SSL access" on page 17. If you want to configure basic SSL, continue with the following steps:
    a) To create a self-signed certificate:
        i) Open *image_path*`/scripts/ISAMGenSSLCert.sh` in a text editor.
        ii) Set the password for the key database with the `KEYFILEPWD` variable.
        iii) Save and close the file.
        iv) Run *image_path*`/scripts/ISAMGenSSLCert.sh`. Replace *image_path* with the path to the script files.

        Note: The self-signed certificate is extracted to `am_key.der`.
    b) To enable SSL with Security Directory Server:
        i) Open *image_path/platform*`/tdsV6.3/idsConfigServerSSL.sh` in a text editor.
        ii) Set the values for the following variables. Values in bold are the typical default values. Use values that are specific and correct for your environment.

```
tdsinstancename=dsrdbm01
port=389
ssl_port=636
serverpwd=
serverlabel=AMLDAP
serverkeywithpath=/am_key.kdb
user_dn=cn=root
password_dn=
```

        Note: The password fields must be set to your passwords.

iii) Save and close the file.

iv) Run *image_path/platform*/tdsV6.3/idsConfigServerSSL.sh. Replace *image_path/platform* with the path to the Security Directory Server installation files.

Installing Security Directory Server with a script (Windows)

Use the script file to automate the installation of Security Directory Server.

Before you begin

The information in this topic is for Directory Server version 6.3. Use the information and links in the tech note for the configuration instructions for Security Directory Server 6.4 and Security Directory Suite 8.0.

About this task

Automated installations can perform unattended silent installations. This task uses the **install_tdsSilent** command.

Procedure

1. Log on to the system with Administrator privileges.
2. Extract the Security Directory Server archive file to a directory with adequate disk space, for example, /tdsV6.3/. If you use a DVD to install Security Directory Server, the files are in the tdsV6.3 directory.
3. Locate the following script files and change the permissions so that you can write to the files:

   ```
   image_path\tds\optionsFile\InstallServer.txt
   image_path\scripts\ISAMConfigTDS.bat
   image_path\scripts\ISAMGenSSLCert.bat
   image_path\Windows\tdsV6.3\idsConfigServerSSL.bat
   ```

   For example:

   a. For each file previously listed, right-click the file and click **Properties**.

   b. Click the **Security** tab.

   c. In the **Name** list box, select the user or group that you want to change.

   d. In the **Permissions** box, select **Write**.

   e. Click **OK**.
4. In the directory, locate the installation program file and the response file.
   - *image_path*\windows*tdsV6.3*\tds\install_tdsSilent.exe
   - *image_path*\windows*tdsV6.3*\tds\optionsFile\InstallServer.txt
5. Update the entries in the InstallServer.txt file with the appropriate values for your installation.

   Use the instructions in the text file. For more information, see the topics about the options files for silent installation in the Security Directory Server IBM Knowledge Center.
6. Save the InstallServer.txt file.
7. Open a command prompt and change to the following directory:

   ```
   image_path\windows\tdsV6.3\tds
   ```
8. Start the installation by running the following command:

   ```
   install_tdsSilent -is:silent -options image_path\optionsFiles\InstallServer.txt
   ```

   where *image_path* is the full path to the optionsFiles directory.
9. Verify the installation by checking the installation log:

   C:\Program Files\IBM\LDAP\V6.3\var\ldapinst.log

10. Create the default instance and suffix:
    a) Open a command prompt.
    b) Change to the following directory:
       ldap_home\idstools
    c) Run the following command:

```
idsdefinst -p passworddn -w passworduser -e encryptseed
```

       where:

       passworddn
       The administration DN password. For example, cn=root password.

       passworduser
       The database owner password. For example, the password for the user ID dsrdbm01.

       encryptseed
       The encryption seed value. This value is used to create is used to generate a set of Advanced Encryption Standard (AES) secret key values. The length must be 12 - 1016 characters.

11. Configure Security Directory Server for Security Verify Access:
    a) Locate the *image_path*\scripts\ISAMConfigTDS.bat file.
    b) Open the file in a text editor.
    c) Set the adminPW to the cn=root password.
    d) Review the other settings in the file. If you used the default values during the installation of Security Directory Server, no further modification is required.
    e) Save and close the ISAMConfigTDS.bat file.
    f) Open a command prompt.
    g) Run *image_path*\scripts\ISAMConfigTDS.bat.
       Replace *image_path* with the path to the script files.
    h) Verify the configuration by checking the configuration log:
       C:\Users\Administrator\ConfigTDSforISAM.log
12. Optional: If you are setting up Suite B and NIST compliance between your user registry and Security Verify Access components, see "Configuring IBM Tivoli Directory Server for SSL access" on page 17. If you want to configure basic SSL, continue with the following steps:
    a) To create a self-signed certificate:
       i) Open *image_path*\scripts\ISAMGenSSLCert.bat in a text editor.
       ii) Set the password for the key database with the KEYFILEPWD variable.
       iii) Save and close the file.
       iv) Run *image_path*\scripts\ISAMGenSSLCert.bat. Replace *image_path* with the path to the script files.

           Note: The self-signed certificate is extracted to am_key.der.
    b) To enable SSL with Security Directory Server:
       i) Open *image_path*\Windows\tdsV6.3\idsConfigServerSSL.bat in a text editor.
       ii) Set the values for the following variables. Values in bold are the typical default values. Use values that are specific and correct for your environment.

```
tdsinstancename=dsrdbm01
port=389
ssl_port=636
serverpwd=
serverlabel=AMLDAP
serverkeywithpath=C:\am_key.kdb
```

```
user_dn=cn=root
password_dn=
```

Note: The password fields must be set to your passwords.

iii) Save and close the file.

iv) Run *image_path*\Windows\tdsV6.3\idsConfigServerSSL.bat. Replace *image_path* with the path to the Security Directory Server installation files.

Installing Security Directory Server with Launchpad (Windows)

Use the Launchpad installation method to install Security Directory Server and its prerequisite software on a computer that is running the Windows operating system.

Before you begin

Complete the following tasks before you set up IBM Security Directory Server:

- Use the information and links in the tech note for the configuration instructions for Security Directory Server 6.4 and Security Directory Suite 8.0. The information in this topic is for Directory Server version 6.3.
- Review the “User registry considerations” on page 3.
- Access the instructions for the "Typical installation path" method in the IBM Security Directory Server version 6.3 IBM Knowledge Center.

About this task

The Launchpad uses a graphical user interface for the step-by-step installation and initial configuration. The Launchpad installs all the prerequisite software, if it is not already installed.

Then, the Launchpad starts the graphical user interface installation for the Security Directory Server component.

This task installs the following components:

- All components that are required by Security Directory Server.
- All the corequisite products that are required by Security Directory Server, if they are not already installed.
  - GSKit
  - DB2®
- The embedded version of WebSphere® Application Server. This software is required by the Web Administration tool, which is installed automatically as part of the "Typical installation path" method.

This task also completes the following configuration:

- Deploys the Web Administration tool.
- Creates a default directory server instance named `dsrdbm01`.
- Creates the operating system user ID named `dsrdbm01` that owns the instance.
- Creates an Administrator DN named `cn=root`.
- Creates a default suffix named `o=sample`.

Procedure

1. Start the Launchpad.
   a) Locate the `launchpad64.exe` file.

   Note: If you are using archive files, ensure that all of them are extracted into the same directory. For example, ensure that the archive files for the IBM Security Verify Access package and the Security Directory Server packages are extracted into the same directory.

b) Double-click the file to start the Launchpad.

2. Select the language that you want to use during the installation and click **OK**.

   The **Launchpad Welcome** window opens.
3. Click **Next**.
4. Select the **IBM Security Directory Server** component.
5. Click **Next**.

   The list on the left displays the component that you selected and any prerequisite software that is required by that component but that is not already installed.
6. Click **Next**.

   The installation pane for the first component that is listed is displayed. An arrow next to a component name on the left indicates that the component is being installed. A check mark next to a component name indicates that the component is installed.
7. If the current component is IBM Global Security Kit, click **Install IBM Global Security Kit** to install it. When it completes, continue with step "8" on page 16.
8. Click **Next**.
9. Respond to the prompts presented during the installation.
10. Click **Next** at the bottom of the Launchpad.

    The installation wizard for Security Directory Server opens.
11. Respond to the prompts presented during the installation.
12. When prompted for the installation type, select **Typical**.
13. Complete the installation by using the "Typical installation path" instructions in the IBM Security Directory Server IBM Knowledge Center.

    For more information, see IBMSecurity Directory Server version 6.3 IBM Knowledge Center.

    Note: Record any passwords that you set during the installation so that you can use them in subsequent installation steps.
14. Complete the following steps when the Security Directory Server Administrator tool opens.

    a. Verify that the default instance is listed in the configuration.

    b. Do not start the instance.

    c. Exit the tool.
15. After Security Directory Server is installed, you are prompted for the `cn=root` password that you provided during the installation.
16. Click **Configure IBM Security Directory Server**.
17. When all installations and configurations are completed, a success or failure message is displayed. Take one of the following actions:

    - If the installation completed successfully, click **Next**.
    - If the installation failed or an error is displayed, review the log file in the default %USERPROFILE% location, such as `C:\Users\Administrator\ConfigTDSforISAM.log`. Make corrections or reinstall Security Directory Server as indicated by the log file.
18. Click **Finish** to close the Launchpad.

What to do next

If you are setting up SSL communication, see "Configuring IBM Tivoli Directory Server for SSL access" on page 17

Configuring IBM Tivoli Directory Server for SSL access

Enable SSL to secure communication between the Tivoli Directory Server and the Security Verify Access components.

Before you begin

Complete the following tasks:

- Install and configure Tivoli Directory Server.
- Install GSKit.
- Use the information and links in the tech note for the configuration instructions for Security Directory Server 6.4 and Security Directory Suite 8.0. The information in this topic is for Directory Server version 6.3.

About this task

The following high-level steps are required to enable SSL support for Tivoli Directory Server for server authentication. See the information for securing directory communications in the Tivoli Directory Server Knowledge Center for the details of each step. These steps assume that you already installed and configured the Tivoli Directory Server.

Procedure

1. Create the key database, associated password stash file, and password on the Tivoli Directory Server system. For example, use the **gsk8capicmd_64** to create a database, stash file, and password.

   ```
   gsk8capicmd_64 -keydb -create -db /key/myldap.kdb -pw passw0rd
     -type cms -stash -empty
   ```

2. If you do not already have a personal certificate or self-signed certificate, do one of the following procedures:

   For a personal certificate:

   a. Request a personal certificate from a certificate authority (CA).

   b. Receive that personal certificate into the key database file.

   c. Add a signer certificate for the certificate authority to the key database file.

   For a self-signed certificate:

   a. Create a self-signed certificate. For example,

   ```
   gsk8capicmd_64 -cert -create -db /key/myldap.kdb -pw serverpwd \
   -sigalg algorithm_id -label serverlabel
   -dn "cn=LDAP_Server,o=sample" -size keysize
   ```

   where:

   db
   Specifies the .kdb file that is the key database.

   pw
   Specifies the password to access the key database.

   sigalg
   Specifies the signing algorithm that is used to sign the message. Acceptable values that correspond to a compliance mode are listed in the following table.

   Note: This setting requires a minimum version of Tivoli Directory Server 6.3.0.17. Skip this setting if you are using an earlier version of Tivoli Directory Server or if your environment does not require a compliance configuration.

Table 2. Compliance values for the keyfile		
Compliance mode	***algorithm_id* value**	***keysize* value**
none	SHA1WithRSA	2048
fips	SHA1WithRSA	2048
sp800-131-transition	SHA256WithRSA	2048
sp800-131-strict	SHA256WithRSA	2048
suite-b-128	SHA256WithECDSA	256
suite-b-192	SHA384WithECDSA	384

label
Specifies the label that is attached to the certificate. The label name is configured in Security Directory Server. Either the label name must match the Security Directory Server configured value, or you must update the name value in Security Directory Server to match the label that you set here.

dn
Indicates an X.500 distinguished name. An example format: `CN=common_name, O=organization, C=country`.

size
The size of the new key pair to be created. This size ranges in value and depends on the key type.

Note: For some algorithms, you can specify a 0 value to use the default key size. This size is typically the minimum size that is considered secure. The following list contains the valid values.

For RSA algorithms:
`512-4096`; key sizes in this range must be selected as NIST SP800-131; 8192 is supported for validation only.

Note: Available key sizes might vary according to security configurations. For example, you cannot generate 512-bit RSA keys in FIPS mode. The default value is 1024.

For EC algorithms:
`224 - 512`

Note: GSKit EC key generation supports P256, P384, and P521 curves only. P521 curve keys use a 512-bit SHA2 hash. The following list contains the default values.

- 256 (SHA256)
- 384 (SHA384)
- 512 (SHA512)

b. Extract the certificate in ASCII format. For example, type:

```
gsk8capicmd_64 -cert -extract -db /key/myldap.kdb -pw serverpwd
-label myldap -format ascii -target myldap.cert
```

In a subsequent configuration task, you import this certificate to the signer section of the key database on all client systems that securely communicate with the server.

Note: A client system is:

- Any Security Verify Access server system.
- Any other system that uses the Tivoli Directory Server client to securely communicate with the Tivoli Directory Server.
- Any system that uses the Security Verify Access Runtime component

3. Configure the Tivoli Directory Server instance to use the certificate in the configuration file.

   Note: Create an `ldif` file with the appropriate configuration values in it to perform this step. For more information about `ldif` files, see the Tivoli Directory Server Knowledge Center. If you do not create an `ldif` file for this step, you must use standard input to enter the configuration.

   a) Create an `ldif` file that contains the following values.

   Use your own value for the values that are shown in italics.

```
dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslAuth
ibm-slapdSslAuth: serverAuth
```

   Note: Use `serverAuth` or the value that is appropriate for your environment. The other valid value is `serverClientAuth`.

```
dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSecurity
ibm-slapdSecurity: SSL
```

   Note: Use SSL or the value that is appropriate for your environment. The valid values are `none`, SSL, `SSlOnly`, TLS, SSLTLS.

```
dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslKeyDatabase
ibm-slapdSslKeyDatabase: /key/myldap.kdb
```

```
dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslCertificate
ibm-slapdSslCertificate: serverlabel

dn: cn=SSL, cn=Configuration
changetype: modify
replace: ibm-slapdSslKeyDatabasepw
ibm-slapdSslKeyDatabasepw: serverpwd
```

   b) Save the file and name it. For example, name it`serverauth.ldif`.

   c) Run the **ldapmodify** command.

```
idsldapmodify -h server.in.ibm.com -p 389 -D cn=root -w root \
-i /home/dsrdbm01/serverauth.ldif
```

   where:

   h hostname
   Specifies the host on which the LDAP server is running.

   p port_number
   Specifies an alternative TCP port where the LDAP server is listening. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

   D binddn
   Use**binddn** to bind to the LDAP directory. **binddn** is a string-represented DN. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authzId string that starts with **u:** or **dn:**.

   Note: `-D binddn -w passwd` does not call bind functions on superuser DNs.

   i filename
   Specifies the entry modification information from an LDIF file instead of from standard input. If an LDIF file is not specified, you must use standard input to specify the update records in LDIF format.

4. Update the compliance type (such as FIPS), if required for your environment.

Note: This step requires a minimum version of Tivoli Directory Server 6.3.0.17. Skip this step if you are using an earlier version of Tivoli Directory Server or if your environment does not require a compliance configuration.

Create an `ldif` file with the appropriate configuration values in it to perform this step. For more information about `ldif` files, see the Tivoli Directory Server Knowledge Center. If you do not create an `ldif` file for this step, you must use standard input to enter the configuration.

a) Choose the compliance mode that you want to use in your environment.

- `none`
- `fips`
- `sp800-131-transition`
- `sp800-131-strict`
- `suite-b-128`
- `suite-b-192`

For descriptions of these compliance modes, see the documentation that came with the Tivoli Directory Server fix pack.

b) Create an `ldif` file that contains the appropriate values for the compliance mode you want to use.

Table 3. Compliance attribute values

Compliance mode	**Values for `cn=Front End, cn=Configuration`**	**Attributes for `cn=SSL, cn=Configuration`**

none	`ibm-slapdSetenv: IBMSLAPD_SECURITY_PROTOCOL= SSLV3,TLS10,TLS11,TLS12`	`ibm-slapdSecurity: SSLTLS`
		`ibm-slapdSslFIPSModeEnabled: false`
		`ibm-slapdSslFIPSProcessingMode: false`
		`ibm-slapdSslCipherSpec: AES`
		`ibm-slapdSslCipherSpec: AES-128`
		`ibm-slapdSslCipherSpec: RC4-128-MD5`
		`ibm-slapdSslCipherSpec: RC4-128-SHA`
		`ibm-slapdSslCipherSpec: TripleDES-168`
		`ibm-slapdSslCipherSpec: DES-56`
		`ibm-slapdSslCipherSpec: RC2-40-MD5`
		`ibm-slapdSslCipherSpec: RC4-40-MD5`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_3DES_EDE_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_GCM_SHA256`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_GCM_SHA384`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA256`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA256`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`
		`ibm-slapdSslCipherSpec: TLS_RSA_WITH_RC4_128_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_RC4_128_SHA`
		`ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_RC4_128_SHA`

Table 3. Compliance attribute values (continued)

Compliance mode	Values for cn=Front End, cn=Configuration	Attributes for cn=SSL, cn=Configuration
fips	ibm-slapdSetenv: IBMSLAPD_SECURITY_PROTOCOL=TLS10,TLS11,TLS12	ibm-slapdSecurity: SSLTLS ibm-slapdSslFIPsProcessingMode: true ibm-slapdSslCipherSpec: AES ibm-slapdSslCipherSpec: AES-128 ibm-slapdSslCipherSpec: TripleDES-168 ibm-slapdSslCipherSpec: TLS_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
sp800-131-transition	ibm-slapdSetenv: IBMSLAPD_SECURITY_PROTOCOL=TLS10,TLS11,TLS12	ibm-slapdSecurity: SSLTLS ibm-slapdSslFIPsProcessingMode: true ibm-slapdSslCipherSpec: AES ibm-slapdSslCipherSpec: AES-128 ibm-slapdSslCipherSpec: TripleDES-168 ibm-slapdSslCipherSpec: TLS_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
sp800-131-strict	ibm-slapdSetenv: IBMSLAPD_SECURITY_PROTOCOL=TLS12 ibm-slapdSetenv: IBMSLAPD_SSL_EXTN_SIGALG=GSK_TLS_SIGALG_RSA_WITH_SHA224,GSK_TLS_SIGALG_RSA_WITH_SHA256,GSK_TLS_SIGALG_RSA_WITH_SHA384,GSK_TLS_SIGALG_RSA_WITH_SHA512,GSK_TLS_SIGALG_ECDSA_WITH_SHA224,GSK_TLS_SIGALG_ECDSA_WITH_SHA256,GSK_TLS_SIGALG_ECDSA_WITH_SHA384,GSK_TLS_SIGALG_ECDSA_WITH_SHA512	ibm-slapdSecurity: SSLTLS ibm-slapdSslFIPsProcessingMode: true ibm-slapdSslCipherSpec: TLS_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_RSA_WITH_AES_256_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ibm-slapdSslCipherSpec: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Table 3. Compliance attribute values (continued)		
Compliance mode	**Values for `cn=Front End, cn=Configuration`**	**Attributes for `cn=SSL, cn=Configuration`**
`suite-b-128`	`ibm-slapdSetenv: IBMSLAPD_SUITEB_MODE=128`	`ibm-slapdSecurity: SSLTLS`
`suite-b-192`	`ibm-slapdSetenv: IBMSLAPD_SUITEB_MODE=192`	`ibm-slapdSecurity: SSLTLS`

c) Save the file and name it.

For example, name it `compmode.ldif`.

d) Run the **ldapmodify** command.

Replace the values in italics with your own values.

```
idsldapmodify -h server.in.ibm.com -p 389 -D cn=root -w root \
-i /home/dsrdbm01/compmode.ldif
```

where:

h hostname
Specifies the host on which the LDAP server is running.

p port_number
Specifies an alternative TCP port where the LDAP server is listening. The default LDAP port is 389. If **-p** is not specified and **-Z** is specified, the default LDAP SSL port 636 is used.

D binddn
Use **binddn** to bind to the LDAP directory.**binddn** is a string-represented DN. When used with **-m DIGEST-MD5**, it specifies the authorization ID. It can be either a DN or an authzId string that starts with **u:** or **dn:**.

Note: `-D binddn -w passwd` does not call bind functions on superuser DNs.

i filename
Specifies the entry modification information from an LDIF file instead of from standard input. If an LDIF file is not specified, you must use standard input to specify the update records in LDIF format.

5. Make a note of the SSL secure port number on this server.

   The default secure port number is 636.

6. Copy the signer certificate and have it available to copy onto the computer on which Security Verify Access components are installed and with which you want to enable SSL communication.

   In a subsequent task, you add this certificate to the key database on that computer.

IBM Tivoli Directory Server for z/OS installation and configuration

Prepare the LDAP server on z/OS for Security Verify Access.

Particular emphasis is given to configuring Security Verify Access against a Tivoli Directory Server for z/OS that is configured to use its native authentication facility. This native authentication facility uses a System Authorization Facility (SAF) registry.

These guidelines assume a new LDAP server instance that is dedicated to the Security Verify Access registry.

Schema file updates

You must update the z/OS schema to support the current version of Security Verify Access.

This step must be done following the application of the `schema.user.ldif` and `schema.IBM.ldif` files that are supplied with the server.

To apply the Security Verify Access schema to the Security Directory Server, use the **`ivrgy_tool`** utility. For instructions, see the Reference topics in the IBM Knowledge Center.

Suffix creation

Security Verify Access requires that you create a suffix that maintains Security Verify Access metadata.

You must add the suffix only once when you first configure the LDAP server. The suffix enables Security Verify Access to easily locate and manage the data. It also secures access to the data, avoiding integrity or corruption problems.

For more information about management domains and creating a location for the metadata, see Chapter 3, “Security Verify Access management domains,” on page 39 and “Management domain location example” on page 39.

If you decide to add suffixes after the Security Verify Access policy server is configured, you must apply the appropriate ACLs to the newly created suffix. You can use the **ivrgy-tool** to apply the ACLs to the new suffix. For more information about the **ivrgy-tool**, see the Reference topics in the IBM Knowledge Center.

Suffix definitions

Security Verify Access processes all defined LDAP suffixes by default.

If suffixes are defined on the LDAP server that must not be used by Security Verify Access, add them to the */access_mgr_install_dir*`/etc/ldap.conf` file by using the `ignore-suffix` keyword when you configure Security Verify Access for LDAP on z/OS.

For example:

```
ignore-suffix = sysplex=UTCPLXJ8
ignore-suffix = "o=Your Company"
ignore-suffix = o=MQuser
```

In this example, the `sysplex=UTCPLXJ8` suffix is used to access the z/OS SDBM (RACF®) database. The LDAP administrator ID used by Security Verify Access during configuration is not a RACF user ID on the z/OS system and does not have the authority to do SDBM searches. If this suffix was not added to the `ignore-suffix` list, Security Verify Access receives a return code `x'32' - LDAP_INSUFFICIENT_ACCESS`, during configuration.

The other suffixes in the list are used by other applications on z/OS and can be ignored by Security Verify Access.

Security Verify Access supports LDAP failover and load balancing for read operations. If you configured a replica server, you can provide the replica host name to Security Verify Access in the `ldap.conf` file, which is installed with Security Verify Access in the `etc` subdirectory.

Native authentication user administration

Native authentication provides the added feature of many-to-one mapping of Security Verify Access users to SAF user IDs.

Most of the existing administrative tasks work similarly with native authentication. Operations such as **`user create`**, **`user show`**, adding a user to an ACL entry or group, and all **`user modify`** commands (except password) work the same as Security Verify Access configured against any other LDAP registry. Users can change their own SAF passwords with the web-based **pkmspasswd** utility.

Multiple users can have the same **ibm-nativeId**, and all bind with the same password. For this reason, prevent many-to-one mapped users from changing the SAF password. Otherwise, there is an increased risk that users might inadvertently lock their peers out of their accounts.

```
pdadmin sec_master> group modify SAFusers add user1
pdadmin sec_master> acl create deny_pkms
pdadmin sec_master> acl modify deny_pkms set group SAFusers T
pdadmin sec_master> acl attach /Webseal/server_name/pkmspasswd deny_pkms
```

There is no administration command ready for immediate use to set the `ibm-nativeId` entry for a user. To that end, the following instructions assist the management of Security Verify Access users with an associated `nativeId`.

The **`user create`** command does not change:

```
pdadmin sec_master> user create user1 cn=user1,o=tivoli,c=us user1 user1 ChangeMe1
pdadmin sec_master> user modify user1 account-valid yes
```

The password (`ChangeMe1`, in this example) is set to the user's `userpassword` entry in LDAP, which has no effect with native authentication enabled. In production environments, use the utility program that is provided with the Security Directory Server for z/OS to remove `userpassword` values from LDAP. This prevents password access if native authentication is inadvertently disabled.

To set the `ibm-nativeId` entry for a user, create an `ldif` file, called a *schema file*, similar to the following:

```
dn: cn=user1,o=tivoli,c=us
changetype: modify
objectclass: ibm-nativeAuthentication
ibm-nativeId: SAF_username
```

You can load the `ldif` file by using the **`ldapmodify`** command on z/OS as follows:

```
ldapmodify -h host_name -p port -D bind_DN
-w bind_pwd -f schema_file
```

Note: To run the **`idsldapmodify`** from an Security Directory Server client on a distributed system, the format of the `ldif` file changes slightly.

```
dn: cn=user1,o=tivoli,c=us
objectclass: inetOrgPerson
objectclass: ibm-nativeAuthentication
ibm-nativeId: SAF_username
```

The SAF command to reset a user's password is as follows:

```
ALTUSER SAF_username PASSWORD(new_password)
```

In addition to resetting the password, the command marks the password as expired, which requires the password to be changed during the next login. If wanted, the `NOEXPIRED` option can be added to the command to prevent that behavior.

Note: The SAF_username must be defined as a z/OS UNIX System Services user. That is, the SAF_username must be defined on z/OS with an OMVS segment. The following line is an example of a SAF command to define SAF_`username` as a UNIX System Services user:

```
altuser SAF_username omvs(home(/u/SAF_username) program(/bin/sh) uid(123456))
```

To use native authentication, you must turn off the `auth-using-compare` stanza entry. To do so, edit the `[ldap]` stanza of the `ivmgrd.conf` and `webseald.conf` file and change the line as follows:

```
auth-using-compare = no
```

By default, authentications to LDAP are made with a compare operation, rather than a bind.

After you configure the IBM Security Directory Server for z/OS for use with Security Verify Access, the next step is to set up the policy server.

Configuring IBM Security Directory Server for z/OS for SSL access

When Security Verify Access and LDAP services are not on the same protected network, enable SSL communication between the LDAP server and the clients that support Security Verify Access. This protocol provides secure and encrypted communications between each server and client. Security Verify Access uses these communications channels as part of the process for making authentication and authorization decisions.

About this task

The following high-level steps are required to enable SSL/TLS support on z/OS. These steps assume that you installed and configured the LDAP directory server, installed z/OS Cryptographic Services System SSL, and set `STEPLIB`, `LPALIB`, or `LINKLIST`.

Procedure

1. Configure the LDAP server to listen for LDAP requests on the SSL port for server authentication and optionally, client authentication. See “Security options in the ibmslapd.conf file” on page 25.
2. Generate the LDAP server private key and server certificate. Mark the certificate as the default in the key database or key ring, or identify the certificate by using its label on the `sslCertificate` option in the configuration file.

   The z/OS LDAP Server can use certificates in a key ring that is managed with the RACF **RACDCERT** command.

   The **gskkyman** utility, which was used in previous releases, also can be used and an example of using that utility to create a key database file can be found in “Creating a key database file” on page 26.
3. Restart the LDAP server.

Security options in the ibmslapd.conf file

You can modify the `ibmslapd.conf` file so that you can configure the options for SSL.

`listen` *ldap_URL*
Specifies, in LDAP URL format, the IP address, or host name and the port number where the LDAP server listens to incoming client requests. This parameter can be specified more than one time in the configuration file.

`sslAuth` `serverAuth` | `serverClientAuth`
Specifies the SSL/TLS authentication method. The **`serverAuth`** method allows the LDAP client to validate the LDAP server on the initial contact between the client and the server. The **`serverAuth`** method is the default.

`sslCertificate` *certificateLabel* | none
Specifies the label of the certificate that is used for server authentication. This option is needed if a default certificate is not set in the key database file or key ring, or if a certificate other than the default one is required. If this option is omitted, the default certificate is used.

`sslCipherSpecs` *string* | `ANY`
Specifies the SSL/TLS cipher specifications that can be accepted from clients.

`sslKeyRingFile` *filename* | *keyring*
Specifies the path and file name of the SSL/TLS key database file or key ring for the server.

`sslKeyRingFilePW` *string*
Specifies the password that protects access to the SSL/TLS key database file.

When a RACF key ring is used instead of a key database file, do not specify this option in the configuration file.

Note: Use of the **sslKeyRingFilePW** configuration option is discouraged. As an alternative, use either the RACF key ring support or the **sslKeyRingPWStashFile** configuration option. This option eliminates this password from the configuration file.

sslKeyRingPWStashFile *filename*
Specifies a file name where the password for the server key database file is stashed. If this option is present, then the password from this stash file overrides the value that is specified for the **sslKeyRingFilePW** configuration option. Use the **gskkyman** utility with the **–s** option to create a key database password stash file.

When a RACF key ring is used instead of a key database file, do not specify this option in the configuration file.

Creating a key database file

Use the **gskkyman** utility so that you can create a key database file.

Procedure

1. Start the **gskkyman** utility from a shell prompt (OMVS or rlogin session) as follows:

```
$ gskkyman
```

2. Enter option 1 to create a new key database file.
3. Type a key database name or accept the default **key.kdb**.
4. Press Enter
5. Create a password to protect the key database.
6. Enter the database password for verification.
7. Type a password expiration interval in days or accept the default (no expiration date).
8. Type a database record length or accept the default **2500**.
   The key database is created and a message is displayed indicating the success or failure of this operation
9. From the **Key Management** menu, select option **6** to create a self-signed server certificate and follow the prompts.
10. After the certificate is created, you must extract this certificate so it can be sent to the LDAP client system and added as a trusted CA certificate. To do so, follow these steps:
    a) Select option **1** to manage keys and certificates.
    b) From the **Key and Certificate List**, enter the label number of the certificate to be exported.
    c) From the **Key and Certificate** menu, enter option 6to export the certificate to a file.
    d) From the **Export File Format** dialog, select the export format. For example, select option **1** to export to `Binary ASN.1 DER`.
    e) Enter the export file name.

Results

The certificate is exported. You can now transfer the exported file to the LDAP client system and add it as a trusted CA certificate. Since the file format of `binary DER` is specified on the export, this same file type must be specified to the **gsk7ikm** utility on the LDAP client system during the **Add** operation.

Microsoft Active Directory Lightweight Directory Service (AD LDS) installation and configuration

You must prepare the AD LDS server for use with Security Verify Access.

Before you install Microsoft Active Directory Lightweight Directory Service, read “Installing and configuring Active Directory Lightweight Directory Service (AD LDS) ” on page 27, which provides a

summary of important Security Verify Access considerations and requirements when you install and configure AD LDS.

For complete download, installation and configuration instructions, see the AD LDS documentation that is provided by Microsoft Corporation.

Installing and configuring Active Directory Lightweight Directory Service (AD LDS)

Install and configure Active Directory Lightweight Directory Service (AD LDS) so that you can use it as a user registry with Security Verify Access.

Procedure

1. Log on to the system by using an account that belongs to the local Administrators group. Use the **Active Directory Lightweight Directory Service Setup Wizard** to configure your AD LDS instance.
2. When you create an AD LDS instance, you must specify an AD LDS instance name that is used to uniquely identify the instance and name the AD LDS service.
3. Specify the ports that are used for both non-SSL and SSL connection types in the AD LDS instance. Make note of the port numbers you specify because they must be entered when you configure Security Verify Access.
4. On the **Application Directory Partition** pane of the **Active Directory Lightweight Directory Service Setup Wizard**, create an application directory partition to contain the user and group definitions that you use.

   Below the directory partition, you can create other **Directory Information Tree** (DIT) entries as needed.
5. On the **File Locations** pane, specify the directories that are used to store the files that are associated with this instance.
6. On the **Service Account Select** pane, select the account that is used to assign permissions to this instance.
7. On the **AD LDS Administrators** pane, select the account that has administrative control of this instance.
8. On the **Importing LDIF Files** pane of the **Active Directory Lightweight Directory Service Setup Wizard**, import the following LDIF files to update the schema that is used by this instance of AD LDS:
   - `MS-InetOrgPerson.LDF`
   - `MS-User.LDF`
   - `MS-UserProxy.LDF`
9. When you finish installing AD LDS, ensure that the installation completed successfully and did not contain any errors. `adamsetup.log` and `adamsetup_loader.log` contain information that can help you troubleshoot AD LDS setup failure.

Installing Security Verify Access with support for Active Directory Lightweight Directory Service (AD LDS)

To use AD LDS with Security Verify Access, you must copy the `tam-adamschema.ldf` file to the AD LDS server. This file can be obtained from the **File Downloads** section of the appliance in the **ISVA** folder.

Configuring the Security Verify Access schema

Security Verify Access defines its own set of LDAP entry types and attributes that it uses to track user, group, and policy information. Add the Security Verify Access schema extensions so that Active Directory Lightweight Directory Service support is enabled.

Before you begin

Before you add Security Verify Access schema extensions, ensure that you defined `inetOrgPerson` and user schema definitions included with AD LDS. If the `inetOrgPerson` and user schema extensions are not added yet, they can also be added by using the **`ldifde.exe`** command-line tool and must be done before you add the Security Verify Access schema.

About this task

These extensions to the basic LDAP server schema must be added to Active Directory Lightweight Directory Service (AD LDS) before you configure Security Verify Access.

After you install AD LDS and configure the AD LDS instance by using the **Active Directory Lightweight Directory Service Setup Wizard**, the Security Verify Access schema extensions can be added to AD LDS by using the **`ldifde.exe`** command-line tool included with AD LDS.

To add inetOrgPerson and user schema extensions, use the following procedure. After you run these commands, the AD LDS schema includes the AD LDS, `inetOrgPerson`, and user `objectclasses` and attribute definitions. If these schema extensions are already added, you can skip this procedure.

Procedure

1. Apply the `tam-adamschema.ldf` schema file on the AD LDS server.

   Note: The file is in the downloads section of the appliance. In the local management interface, navigate to **System** > **Secure Settings** > **File Downloads** > **ISVA**.
2. Click **Start** > **All Programs** > **Accessories**.
3. Right-click **Command Prompt**.
4. Change to the directory that houses the `ldf` files for AD LDS. The path is similar to the following line:

   ```
   C:\Windows\winsxs\amd64_microsoft-windows-d..services-adam-setup
   _31bf3856ad364e35_6.1.7600.16385_none_981a296d97d2c90a
   ```
5. Click **Run as administrator**.
6. At the command prompt, type the following command and then press Enter:

   ```
   ldifde –i –f ms-inetorgperson.ldf –s servername:portnumber –k –j . –c
       “CN=Schema,CN=Configuration,DC=X” #schemaNamingContext
   ```

   where *servername* represents the workstation name and *portnumber* is the LDAP connection port of your AD LDS instance. If AD LDS is running on your local workstation, you can also use `localhost` as the workstation name.
7. Type the following command, and then press Enter:

   ```
   ldifde –i –f ms-user.ldf –s servername:portnumber –k –j . –c
       “CN=Schema,CN=Configuration,DC=X” #schemaNamingContext
   ```

   where *servername* represents the workstation name and *portnumber* is the LDAP connection port of your AD LDS instance. If AD LDS is running on your local workstation, you can also use `localhost` as the workstation name.
8. After you ensured that the AD LDS schema includes the `inetOrgPerson` and user definitions, add the Security Verify Access schema extensions:
   a) Click **Start** > **All Programs** > **Accessories**.
   b) Right-click **Command Prompt**.

c) Click **Run as administrator**.

d) Change to the directory that contains the `tam-adamschema.ldf` file.

e) At the command prompt, type the following command and then press Enter:

```
ldifde –i –e –f tam-adamschema.ldf –s servername:portnumber –k –j . –c
    “CN=Schema,CN=Configuration” #schemaNamingContext
```

where *servername* represents the workstation name and *portnumber* is the LDAP connection port of your AD LDS instance. If AD LDS is running on your local workstation, you can also use `localhost` as the workstation name. The `tam-adamschema.ldf` file is included in the **File Downloads** area of the Security Verify Access appliance.

Management domain data location

The user registry creates and stores metadata that tracks information about the Security Verify Access management domain. You must specify the location for the metadata storage.

The management domain is created when the Security Verify Access policy server is configured. The management domain is the initial security domain.

During policy server configuration, the administrator specifies the name of the management domain or uses the default name of `Default`.

The administrator also specifies the location in the registry where this metadata is stored by specifying the management domain location DN. The location that is specified must exist in the user registry. If the administrator chooses to use the default management domain location, the information is maintained in specific Active Directory Lightweight Directory Service (AD LDS) partition, which must be called

```
secAuthority=management_domain_name
```

where *management_domain_name* is the management domain name specified. For example, if the default management domain name is used, the partition would be called `secAuthority=Default`. If the administrator does not use the default location and specifies the management domain location DN, any existing location within the AD LDS registry can be used if it is a container object.

Note: You must choose a location DN within the same directory partition where the user and group information is stored. AD LDS requires the policy server to exist in the same directory partition as the user and group information.

The policy server cannot maintain user and group information that is outside of the AD LDS directory partition where the policy server itself is defined.

For this reason, do not use the default management location during policy server configuration when AD LDS is used as the Security Verify Access registry. Instead, choose a management domain location within the AD LDS partition in which you want to maintain the user and groups that reflects your enterprise structure.

Attention: If you chose the default management location during policy server configuration, the option to permanently remove domain information from registry deletes all data in the AD LDS partition of the default domain management location, including registry-specific data, when you unconfigure the Security Verify Access. To retain registry-specific data, choose the management domain location in the AD LDS partition in which you want to maintain users and groups. The default management location is the location for Security Verify Access metadata.

Configuring a Security Verify Access directory partition

By default, Security Verify Access maintains its metadata information in a specific Active Directory Lightweight Directory Service (AD LDS) directory partition that is also known as a naming context or suffix. This default Security Verify Access metadata directory partition is called `secAuthority=Default`. To create the default Security Verify Access metadata directory partition, use the AD LDS administration tool **`ldp.exe`**.

About this task

You must create the partition after the Security Verify Access schema extensions are added to AD LDS and before the Security Verify Access Policy Server is configured. For more information about adding schema extensions, see “Configuring the Security Verify Access schema ” on page 28.

The **ldp.exe** tool is installed as part of the AD LDS administration tool set. To use the**ldp.exe** tool, you must connect and bind to the AD LDS instance by using the following procedure.

Alternatively, you can choose a non-default Management Domain name and location DN. The Management Domain name must be unique within the LDAP server and the location DN must exist.

Note: You must choose a location DN within the same directory partition where you store user and group information. This step is required because AD LDS requires that the policy server must exist in the same directory partition in which user and group information is maintained. The policy server cannot maintain user and group information outside the directory partition in which the policy server itself is defined.

Procedure

1. Connect to the AD LDS instance:
   a) At a command prompt, type `ldp` and then press **ENTER**. The **ldp** window is displayed.
   b) On the **Connection** menu, click **Connect....**
   c) In the **Server** field, type the host or DNS name of the system that runs AD LDS. When the AD LDS instance is running locally, you can also type `localhost` for this field value.
   d) In the **Port** field, type the LDAP or SSL port number for the AD LDS instance to which you want to connect. Then, click **OK**. The **ldp** tool connects to the AD LDS instance and displays progress information that is obtained from the root DSE in the pane on the right side of the window.
2. Bind to the AD LDS instance:
   a) From the **Connection** menu, select **Bind...**
   b) To bind by using the credentials that you are logged on with, click **Bind as currently logged on user**.
   c) When you are finished specifying bind options, click **OK**. The **ldp** tool binds the AD LDS instance by using the method and credentials specified.
3. Add the children.
   a) From the **Browse** menu, select **Add child**.
   b) In the **Dn** field, type `secAuthority=Default` as the distinguished name for the new directory partition.
   c) In the **Edit Entry** field, type the following and then click **ENTER**.
      - In the **Attribute** field, type `ObjectClass`.
      - In the **Values** field, type `secAuthorityInfo`.
   d) In the **Edit Entry** field, type the following and then click **ENTER**.
      - In the **Attribute** field, type `secAuthority`.
      - In the **Values** field, type `Default`.
   e) In the **Edit Entry** field, type the following and then click **ENTER**.
      - In the **Attribute** field, type `version`.
      - In the **Values** field, type `8.0`.
   f) In the **Edit Entry** field, type the following and then click **ENTER**.
      - In the **Attribute** field, type `cn`
      - In the **Values** field, type `secAuthority`
   g) In the **Edit Entry** field, type the following and then click **ENTER**.

- In the **Attribute** field, type `instanceType`.
- In the **Values** field, type 5.

The set of attributes and values appear in the Entry List pane.

h) Ensure the **Synchronous** option is selected and click **Run**. This step adds the Security Verify Access metadata directory partition to the AD LDS instance. To verify that the partition is properly added, you can use the AD LDS ADSI Edit tool to connect to and view the partition.

Adding an administrator to the Security Verify Access metadata directory partition

After you add a Security Verify Access schema to the Active Directory Lightweight Directory Service (AD LDS) instance and specified the Security Verify Access metadata directory location, you must add an AD LDS user administrator for the Security Verify Access metadata directory partition.

About this task

The AD LDS user has administrative authority for the Security Verify Access metadata directory partition and is specified as the LDAP administrator during Security Verify Access configuration. The following example assumes that you accepted the default management domain and location. If you specified a different domain name or location, add the AD LDS user administrator to the AD LDS partition you specified.

Procedure

1. Create the AD LDS LDAP administrator:
   a) Start the ADSI Edit program (`Adsiedit.msc`).
   b) On the **Action** menu, click **Connect To**
   c) In the **Connection name** field, you can type a label under which this connection appears in the console tree of AD LDS ADSI Edit. For this connection, type: `secAuthority`.
   d) Under **Connection Point**, enter "`secAuthority=Default`" in the **Select or type a Distinguished Name or Naming Context** section. If you use a different directory partition, select that partition. This example assumes the default partition.
   e) Under **Computer**, enter the server name and port for the AD LDS instance in the **Select or type a domain or server** section. If the AD LDS instance is on the local system, you can use localhost as the server name.
   f) Click **OK**. The term, `secAuthority`, must now appear in the console tree.
2. Select user attributes:
   a) Expand the secAuthority tree by double-clicking **secAuthority** and then double-click **SECAUTHORITY=DEFAULT**.
   b) Highlight and right-click the **SECAUTHORITY=DEFAULT** container, point to **New,** and then click **Object...**
   c) Under **Select a class**, click **user**.
   d) Click **Next**.
   e) For the value of the `cn` attribute, type the common name for the administrator you want to create. For example, type `tam`.
   f) Click **Next**.
   g) Click **More Attributes**.
   h) Select and set the following properties:

msDS-UserDontExpirePassword
Set to **True**. This setting prevents the default password expiration time policy from applying to this administrator. If you would prefer that the password policy applies to this administrator, then this property can be left unset.

msDS-UserAccountDisabled
Set to **False**. This setting enables the instance that you created.

i) Click **OK**.

j) No additional attributes are required but if you want to set more attributes, click **More Attributes**, select the attributes that you want to set and enter the values. When you are finished, click **Finish**. The user is created with a Distinguished Name (DN) of `cn=tam,secAuthority=Default`.

k) To set the administrator password, highlight and then right-click the user that you created. Select **Reset password...**

l) In the "Reset Password" pane, enter and confirm the password that you want to use. When finished, click **OK**. Remember the user DN and password that you create because these details are specified as the LDAP Administrator DN and password when Security Verify Access is configured.

3. Add the user to the Administrators group for the partition:
   a) Within the `SECAUTHORITY=DEFAULT` directory partition, three containers are called `CN=LostAndFound`, `CN=NTDSQuotas`, and `CN=Roles`.
      i) Highlight the **CN=Roles** container by single clicking it. In the details pane on the right side of the AD LDS ADSI Edit tool, the groups within the Roles container are displayed.
      ii) Highlight the **CN=Administrators** group by clicking it.
      iii) Right-click on the **CN=Administrators** group and select **Properties**. The **CN=Administrators Properties** page is displayed.
   b) Under **Attributes**, scroll down and select the **member** attribute.
   c) Click **Edit**.
   d) Click **Add DN**. Type the distinguished name of the administrator user that you created into the **DN** field.
   e) Click **OK**. The administrator user is added to the Administrators group and is displayed as a member.
   f) Click **OK** to complete the membership update. Click **OK**.

Allowing anonymous bind

In order for Security Verify Access to be configured with Active Directory Lightweight Directory Service (AD LDS), AD LDS must be configured to allow anonymous bind.

About this task

By default, AD LDS does not allow anonymous bind. Security Verify Access configuration, however, uses anonymous bind to check on the validity of the configured LDAP host name, port, and SSL parameters.

If you want to disable anonymous bind during normal operation, you can reset the option on the AD LDS server after configuration is complete.

Procedure

1. Start the ADSI Edit program `Adsiedit.msc`.
2. On the **Action** menu, click **Connect To**.
3. In the **Connection name** field, you can type a label under which this connection appears in the console tree of AD LDS ADSI Edit. For this connection, type: `Configuration`.
4. Under **Connection Point**, select **well known Naming Context** and choose **Configuration** from the list.

5. Under **Computer**, enter the server name and port for the AD LDS instance in the **Select or type a domain or server** section. If the AD LDS instance is on the local system, you can use localhost as the server name.
6. Click **OK**. The term, `Configuration`, must now appear in the console tree.
7. Expand the Configuration subtree by double-clicking **Configuration**.
8. Double-click **CN=Configuration,CN=***GUID*, where *GUID* was generated when the configuration of the AD LDS instance was performed.
9. Double-click the **CN=Services** folder to expand it, and then double-click **CN=Windows NT**.
10. Highlight and right-click **CN=Directory Service** and click **Properties**.
11. Click **dsHeuristics**.
12. Click **Edit**.
13. Edit the value. Modify the seventh character (counting from the left) to 2. The value must be similar to `0000002001001` in the String Attribute Editor.
14. Click **OK**.
15. Click **OK**. Anonymous bind is now allowed.

What to do next

If you are setting up SSL communication, see "Configuring Active Directory Lightweight Directory Service (AD LDS) to use SSL" on page 33.

Configuring Active Directory Lightweight Directory Service (AD LDS) to use SSL

Enable SSL to secure communication between the Active Directory Lightweight Directory Service and the Security Verify Access components.

Before you begin

Install and configure Active Directory Lightweight Directory Service, including the Internet Information Service and the Web Management Service.

About this task

SSL encrypts the data that is transmitted between the Security Verify Access services and Active Directory Lightweight Directory Service. Consider enabling SSL to protect information such as user passwords and private data. SSL is not required for Security Verify Access to operate.

The following task summarizes the steps that are required for enabling SSL communications.

Note: For details about enabling SSL on Active Directory Lightweight Directory Service, see the Microsoft documentation for Windows 2008 and Active Directory Lightweight Directory Service.

Procedure

1. Create a certificate that contains the public and private key on the computer where Active Directory Lightweight Directory Service is installed.
2. Export the certificate with its private key.
3. Locate the exported key file, double-click it, and install the certificate in all the folders in the Personal and Trusted Authorities folder.
4. Using the **mmc** console, import this certificate into the Personal and Trusted Root certificate authorities folders for the Active Directory Lightweight Directory Service instance.
5. Change the file permissions of the private keys in the certificate.

   See the Microsoft documentation for details.
6. Restart the Active Directory Lightweight Directory Service instance.

7. Using the **mmc** console, export the **Issue by** certificate of the certificate that is created in Step 1 (do not export the private key) from the *AD_LDS_instance*\Personal folder and save the certificate as a `.cer` file.
8. Import the `.cer` file into an SSL certificate database on the appliance. Use this certificate to configure Security Verify Access with SSL enabled.

Installing and configuring the Oracle Directory Server Enterprise Edition (ODSEE)

You can use a supported version of Oracle Directory Server as the user registry for Security Verify Access.

Before you begin

Review the "User registry considerations" on page 3 before you configure the Oracle Directory Server in your environment:

About this task

Complete the basic server installation and configuration as described in the Oracle Directory Server Enterprise Edition product documentation. For example, for Oracle Directory Server Enterprise Edition version 11, see:

- Installation Guide
- Administration Guide

Then, use the same documentation to create a suffix for Security Verify Access.

Procedure

1. Create the management domain location that maintains Security Verify Access data.

   Use the suffix DN of the location; for example: `secAuthority=Default`.

   The name must be in the relative distinguished name (DN) format and consist of one attribute-value pair. If multiple attribute-value pairs, separate the pairs by commas. The default location is `secAuthority=Default`.

   For more information about management domains, and creating a location for the metadata, see:

   - Chapter 3, "Security Verify Access management domains," on page 39
   - "Management domain location example" on page 39
2. Change the name of the database when you create a suffix.

   

   Attention: Do not accept the default value for the database name when you create a suffix. By default, the location of database files for this suffix is chosen automatically by the server. By default, the suffix maintains only the system indexes. No attributes are encrypted, and replication is not configured. If you accept the default value, the Oracle Directory Server Enterprise Edition stores the suffix under the **Default** database name. Your data is removed when the Oracle Directory Server Enterprise Edition is restarted.
3. Ensure that the suffix was created.

   If you chose to create a suffix to maintain user and group data, follow this procedure again to create another suffix either in the default database or in a new database. For example, you might create a suffix that are named `o=ibm,c=us` in the same database.
4. Complete the appropriate action:

   - If you did not add any suffixes other than the management domain location, configuration is complete. A directory entry for the management domain location is automatically added when the policy server is configured.

- If you added suffixes other than the management domain location, create directory entries for each new suffix.

5. If you want to enable SSL communication between the Directory Server and Security Verify Access, continue with the remaining steps:
   a) Start the instance of the Oracle Directory Server Enterprise Edition.
   b) Obtain a certificate for the instance and store it in the key database. The certificate can be issued by a certificate authority (CA) or it can be self-signed. The certificate includes a server certificate and a private key. Use the methods that are described in the Oracle Directory Server Enterprise Edition documentation.
   c) Make a note of the secure SSL port number on the server. The default port number is 636.
   d) Obtain the signer certificate.

      Note: If the certificate is issued by a CA, the server certificate includes a signer certificate. If the certificate is self-signed, the server certificate acts as the signer certificate.
   e) Copy the signer certificate to a temporary directory on the computer where Security Verify Access components are installed and with which you want to enable SSL communication.

What to do next

After you set up the Directory Server for use with Security Verify Access, you can set up the policy server. Use the following values in your configuration:

- Value of LDAP administrator ID for the Oracle Directory Server is `cn=Directory Manager`. The default value for this attribute is `cn=root`, however, it is not appropriate for the Oracle Directory Server.
- Value of LDAP management domain location DN for the Oracle Directory Server is a suffix (for example, `dc=ibm,dc=isva`) created under the directory instance. The default value for this attribute is blank and it is not appropriate for the Oracle Directory Server.

Installing and configuring the OpenLDAP Server

You can use a supported version of the OpenLDAP Server as the user registry for Security Verify Access.

Before you begin

Review the User registry considerations before you configure the OpenLDAP Server in your environment.

About this task

Complete the basic server installation and configuration as described in the OpenLDAP server product documentation.

Note: The following instructions are for the OpenLDAP '`cn=config`' configuration format rather than the legacy '`slapd.conf`' configuration style.

Procedure

1. **Apply the Security Verify Access schema to the OpenLDAP server:**

   The schema for the Security Verify Access data must be applied to the OpenLDAP server. In order to achieve this:

   a. Obtain the `verify-access-openldap-schema.ldif` file from the `isva` directory of the file downloads section of a running Security Verify Access appliance. This file should be copied to the OpenLDAP server.
   b. On the OpenLDAP server apply the schema `ldif` file to the running server. This can be achieved by using a command similar to the following:

   ```
   ldapadd -c -Y EXTERNAL -Q -H ldapi:/// -f verify-access-openldap-schema.ldif
   ```

2. **Create the suffix for the Security Verify Access data:**

A suffix must be available to house the Security Verify Access data (otherwise known as the `secAuthority` data). This suffix can be added to an existing database by adding a new `olcSuffix` entry to an existing database definition, or an existing suffix can be specified during configuration.

A preferable alternative is to house the suffix in a new '`olcDatabase`'. In order to achieve this:

a. Create a new `olcDatabase` LDIF definition, similar to the following but customised to match your OpenLDAP server environment:

```
dn: olcDatabase={2}mdb,cn=config
changetype: add
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {2}mdb
olcDbDirectory: /var/lib/ldap.secAuthority
olcSuffix: secAuthority=Default
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by
group=cn=SecurityGroup,secAuthority=Default write by dn=cn=root,secAuthority=Default
write by group=cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default read by
group=cn=ivacld-servers,cn=SecurityGroups,secAuthority=Default read by anonymous auth by
* none
olcAccess: {1}to * by group=cn=SecurityGroup,secAuthority=Default
write by dn=cn=root,secAuthority=Default write by group=cn=remote-
acl-users,cn=SecurityGroups,secAuthority=Default read by group=cn=ivacld-
servers,cn=SecurityGroups,secAuthority=Default read by self read by users auth by
anonymous auth
olcLastMod: TRUE
olcRootDN: cn=root,secAuthority=Default
olcRootPW: admin
olcDbCheckpoint: 512 30
olcDbIndex: objectClass                       eq,pres
olcDbIndex: ou,cn,mail,surname,givenname      eq,pres,sub
olcDbIndex: uidNumber,gidNumber,loginShell    eq,pres
olcDbIndex: uid,memberUid                     eq,pres,sub
olcDbIndex: nisMapName,nisMapEntry            eq,pres,sub
olcDbIndex: uniqueMember,secUUID,secAuthority,secDN,secDomainID,member,principalName eq
```

b. Ensure that the directory referenced by the '`olcDbDirectory`' entry has been created and has the appropriate permissions.

c. Apply the LDIF file to the running LDAP server by running a command similar to the following on the LDAP server:

```
ldapadd -c -Y EXTERNAL -Q -H ldapi:/// -f verify-access-suffix.ldif
```

3. **Update permissions in existing suffixes for the Security Verify Access Groups:**

   In order for Security Verify Access to be able to manage standard LDAP users' the permissions for each '`olcDatabase`' must be adjusted in the LDAP configuration. In particular the following '`olcAccess`' permissions are required:

Entry/Attributes	Permission	Entities
userPassword shadowLastChange	Read	group=cn=remote-acl-users,cn=SecurityGroups,<secAuthority-suffix> group=cn=ivacld-servers,cn=SecurityGroups,<secAuthority-suffix>
	Write	dn=<admin-dn> group=cn=SecurityGroup,<secAuthority-suffix>
*	Read	group=cn=remote-acl-users,cn=SecurityGroups,<secAuthority-suffix> group=cn=ivacld-servers,cn=SecurityGroups,<secAuthority-suffix>
	Write	dn=<admin-dn> group=cn=SecurityGroup,<secAuthority-suffix>

The '<admin-dn>' should match the 'olcRootDN' entry in the 'olcDatabase' which is housing the Security Verify Access data. This is the DN which should be specified when configuring the Verify Access runtime component.

The '<secAuthority-suffix>' will be of the format 'secAuthority=<verify-access-domain>{,<verify-access-suffix>}, where:

- <verify-access-domain> is the domain name used when configuring the runtime component (default: 'Default');
- <verify-access-suffix> is the suffix in which the secAuthority data will reside (if no suffix is specified the data will reside in the 'secAuthority=<verify-access-domain>' suffix).

In order to update the permissions:

a. Create a new LDIF definition, similar to the following but customised to match your OpenLDAP server environment:

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
delete: olcAccess

add: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange by self write
by dn="cn=admin,dc=ibm,dc=com" write by dn=cn=root,secAuthority=Default
write by group=cn=SecurityGroup,secAuthority=Default write by
group=cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default read by group=cn=ivacld-
servers,cn=SecurityGroups,secAuthority=Default read by anonymous auth by * none
olcAccess: to * by self write by dn="cn=admin,dc=ibm,dc=com" write by
dn=cn=root,secAuthority=Default write by group=cn=SecurityGroup,secAuthority=Default
write  by group=cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default read by
group=cn=ivacld-servers,cn=SecurityGroups,secAuthority=Default read by * none
```

b. Apply the LDIF file to the running LDAP server by running a command similar to the following on the LDAP server:

```
ldapadd -c -Y EXTERNAL -Q -H ldapi:/// -f verify-access-permissions.ldif
```

What to do next

After you set up the Directory Server for use with Security Verify Access, you can configure the runtime component. Use the following values in your configuration:

- The LDAP administrator ID for the OpenLDAP Server is the 'olcRootDN' entry for the 'olcDatabase' which is housing the Security Verify Access data, for example: 'cn=root,secAuthority=Default'.

Note: If you experience issues when connecting to the OpenLDAP Server over TLS you might need to modify the OpenLDAP server configuration, removing the 'olcTLSCipherSuite' entry.

Chapter 3. Security Verify Access management domains

If you use LDAP as your user registry, Security Verify Access provides for one or more administrative domains. A domain consists of all the users, groups, and resources that require protection along with the associated security policy used to protect those resources.

Depending on the installed resource managers, resources can be any physical or logical entity, including objects such as files, directories, web pages, printer and network services, and message queues. Any security policy that is implemented in a domain affects only the objects in that domain. Users with authority to complete tasks in one domain do not necessarily have the authority to complete those tasks in other domains.

The initial domain in an LDAP registry is called the *management domain* and is created when the policy server is configured. During policy server configuration, you are prompted for the management domain name and the management domain location Distinguished Name (DN) within the LDAP Directory Information Tree (DIT) on the LDAP server where the information about the domain is maintained.

If the management domain location is not specified, the management domain location is assumed to be a stand-alone suffix on the LDAP server. Whether you use the default location or specify a different location in the LDAP DIT, the location that is specified for the management domain must exist unless the user registry is Novell eDirectory. For Novell eDirectory, if you do not specify the management domain location, Security Verify Access uses the root location as the management domain location. The root location is a domain location that does not have a suffix. If you enter a specific location for the management domain, ensure that the location you are specifying exists.

When a Security Verify Access domain is created, including the initial management domain, an entry is created in the LDAP server that is called a `secAuthorityInfo` object. This object represents the Security Verify Access domain and is named according to the secAuthority attribute with the name of the domain as its value; for example: `secAuthority=<domain_name>`.

If you do not provide a different name, the default name of the management domain is `Default`, making the `secAuthorityInfo` object name `secAuthority=Default`.

Management domain location example

If you want to specify a nondefault location for the management domain, you can use any location within the LDAP DIT.

For example, if the LDAP server is configured with a suffix of `c=us`, and the administrator specifies the management domain location DN as `ou=austin,o=ibm,c=us`, this object might be created by using a file that contains the following LDIF:

```
dn: c=us
objectClass: top
objectClass: country
c: US

dn: o=ibm,c=us
objectClass: top
objectClass: organization
o: IBM

dn: ou=austin,o=ibm,c=us
objectClass: top
objectClass: organizationalunit
ou: Austin
```

The object might then be created by using the **idsldapadd** command-line utility as follows:

```
idsldapadd –h <ldap_hostname> -p <ldap_port> -D <ldap_admin_DN>
   -w <ldap_admin_pwd> -v –f example_DIT
```

where:

- *ldap_hostname* is the host name of the LDAP server.
- *ldap_port* is the port of the LDAP server.
- *ldap_admin_DN* is the Distinguished Name of the LDAP server administrator.
- *ldap_admin_pwd* is the password of the LDAP server administrator.
- *example_DIT* is the name of the file that contains the LDIF.

Modify this example for the specific LDAP namespace appropriate for your organization.

After the LDAP object is created, you can specify it as the management domain location DN during policy server configuration.

Note:

If the following conditions exist, a WebSEAL instance cannot change user passwords because of the absence of ACL settings that are required to search domain locations:

- You configured the policy server in a nondefault location that is a location other than `secAuthority=Default`.
- You create Security Verify Access subdomains under the new location.
- You configured a WebSEAL instance in any of the new subdomains.

If you configure the policy server in a nondefault location and find that these other conditions exist, see the Troubleshooting topics in the IBM Knowledge Center for information about setting the correct ACL.

Management domain location for an Active Directory Lightweight Directory Service (AD LDS) registry

If Active Directory Lightweight Directory Service (AD LDS) is being used as the LDAP registry, you must choose a location DN within the same directory partition where you want to store user and group information.

AD LDS has a restriction that the policy server must exist in the same directory partition in which user and group information is maintained. The policy server cannot maintain user and group information outside the directory partition in which the policy server itself is defined.

Chapter 4. Security Directory Server proxy environment setup

A Security Directory Server proxy is a special type of IBM Security Directory Server that provides request routing, load balancing, fail over, distributed authentication and support for distributed/membership groups and partitioning of containers.

Attention: IBM Security Verify Access customers who want to use the Security Directory Server proxy server must purchase a separate Security Directory Server entitlement. The version of Security Directory Server that is part of the Security Verify Access package does not allow IBM Security Verify Access customer-use of the Security Directory Server proxy server.

If you have the appropriate entitlement, see the proxy server instructions in the IBM Security Directory Server Administration Guide.

Then, return to this document for instructions about setting up the proxy server to work with IBM Security Verify Access.

Security Verify Access stores its metadata within a required suffix called `secAuthority=Default`. Metadata includes information that is used to track user and group status information specific to Security Verify Access. When you use a proxy, the `secAuthority=Default` object itself cannot be modified by using the proxy because the object at a proxy partition split point cannot be modified through the proxy. Therefore, Security Verify Access cannot be configured directly through the proxy because Security Verify Access must modify the `secAuthority=Default` object during configuration.

In a proxy environment, the administrator must decide on the back-end server that the `secAuthority=Default` subtree is hosted and set up that back-end server and the proxy partition information to reflect that topology. This example configures `Server A` to host the `secAuthority=Default` subtree.

Data under a proxy partition split point (for example, `o=ibm,c=us`) is hashed to determine which back-end server has the subtree. In this example, `Proxy` is configured to hash RDN values immediately after `o=ibm,c=us` among two servers. This example also means the RDN values more than 1 away from `o=ibm,c=us` will map to the same server as values immediately after `o=ibm,c=us`. For this reason, it is more advantageous to configure the proxy with a single partition for the `secAuthority=Default` suffix.

If you want to distribute the Security Verify Access metadata within the `secAuthority=Default` suffix among multiple back-end servers, it is best to split the partition below the `cn=Users,secAuthority=Default` container. Entries are made on behalf of each user who is defined, below the `cn=Users,secAuthority=Default` container and therefore splitting this user information can help distribute the data more evenly across the back-end servers. This example does not distribute the data but instead maintain the entire `secAuthority=Default` subtree within `Server A`.

Adding the Security Verify Access suffix to the proxy

For the proxy to work with Security Verify Access, you must configure the `secAuthority=Default` suffix.

Procedure

1. Log in to `Server A` as the local LDAP administrator.
   For example, `cn=root`.
2. Select **Server administration** > ➔ > **Manage server properties**. Select the **Suffixes** property.
3. In the **Suffix DN** field, type `secAuthority=Default`.
4. Click **Add**.
5. When you are finished, click **Apply** to save your changes without exiting, or click **OK** to apply your changes and exit.

6. The suffix is available until the server is restarted. In the navigation pane, select **Server administration** and then select **Start/stop/restart server**.
7. Ensure the **Start/restart in configuration only mode** check box is not selected.
8. Click **Restart**.
9. After a message is displayed that the restart request was sent, go to **Server administration** and check the status of the server. Wait until the server restarts successfully and is running before you continue.
10. Log in to `Proxy` as the local LDAP administrator.
    For example, `cn=root`.
11. From the navigation pane, expand **Proxy administration**.
12. On the Proxy administration page, click **Manage proxy properties**.
13. In the **Suffix DN** field, type `secAuthority=Default`.
14. Click **Add**.
15. Click **OK** to save your changes and return to the **Introduction** window.
16. From the navigation pane, click **Proxy administration** and then click **Manage partition bases**.
17. From the **Manage partition bases** menu, click **Add**.
18. In the **Split Name** field, type: `Split 1`
19. In the **Partition base DN** field, type: `secAuthority=Default`
20. In the **Number of partitions** field, type: `1`
21. In the **Partition bases** table, select **secAuthority=Default**.
22. Click **View servers** and then verify that `secAuthority=Default` is displayed in the **Partition base DN** field.
23. In the **Back-end directory servers for partition base** table, click **Add**.
24. From the **Add Back-end directory server** menu, click **Back-end directory server** > **➔** > **Server A**.
25. Ensure that 1 is displayed in the **Partition index** field.
26. Click **OK**.
27. When you are finished, click **Close**.
28. Restart `Proxy` for the changes to take effect.

Security Verify Access configuration with the proxy

After the Security Directory Server proxy server and back-end servers are configured with the Directory Information Tree (DIT) partitioning setup, you can configureSecurity Verify Access to use the proxy. The proxy server provides a unified view of the directory and shields the LDAP application (Security Verify Access for example) from having to be aware of the DIT partitioning.

When configured to use the Security Directory Server proxy server, Security Verify Access is only aware of the proxy and performs all operations through the proxy, as if it represented the entire DIT namespace.

To provide failover support, multiple Security Directory Server proxy servers can also be configured. For information about configuring multiple Security Directory Server proxy servers to provide failover support, see the IBM Tivoli Directory Server Administration Guide.

When you configure multiple proxy servers to provide failover support, Security Verify Access must be configured to treat each of the proxy servers as a directory server replica. The example scenario that is described here assumes a single proxy.

Because Security Verify Access cannot be configured directly to the Security Directory Server proxy server, Security Verify Access must first be configured to the back-end server that hosts the `secAuthority=Default` subtree. When you configure the Security Verify Access Runtime component for use with this back-end server, select **LDAP** as the registry type. When the **pdconfig** utility requests the **LDAP hostname**, type the host name and **LDAP port number** of `Server A` (the back-end server that hosts the `secAuthority=Default` subtree); do not type the host name of the Security Directory Server proxy server (`Proxy`).

Configure SSL information for setting up an SSL connection with `Server A`, if SSL is to be used. When you use SSL, `Proxy` needs to be configured with a server certificate that is generated by the same certificate authority (CA) that was used to create the server certificate for `Server A`. Specify the LDAP DN (for example `cn=root`) and the LDAP administrator password for `Server A`. After the Security Verify Access policy server is configured successfully to the back-end server (`Server A`), you can then retarget the Security Verify Access policy server system to the Security Directory Server proxy server. Exit the **pdconfig** utility.

Redirecting the policy server to the proxy

To retarget the Security Verify Access policy server system to the proxy, edit the policy server `ldap.conf` and `pd.conf` configuration files.

Procedure

1. Log in the local management interface.
2. From the top menu, select **Web** > **Manage** > **Runtime Component**.
3. Click **Manage** > **Configuration Files**.
4. Select **ldap.conf**.
5. Specify the host name and port number of the proxy server.
   - `ldap host` *proxy_hostname*
   - `ldap port` *proxy_port*
6. Click **Save**. If you do not want to save the changes, click **Cancel**. If you want to revert to the previous version of the configuration file, click **Revert**.
7. Click **Manage** > **Configuration Files**.
8. Select **pd.conf**.
9. Specify the host name and port number of the proxy server.
   - `pdrte user-reg-server` *proxy_hostname*
   - `pdrte user-reg-host` *proxy_hostname*
   - `pdrte user-reg-hostport`*proxy_port*
10. Click **Save**. If you do not want to save the changes, click **Cancel**. If you want to revert to the previous version of the configuration file, click **Revert**.

Setting access controls for the proxy

Access control lists (ACLs) cannot be managed from the Security Directory Server proxy server. When a proxy server is used, it is the back-end server that enforces access control. The LDAP administrator must ensure that the proper ACLs are created on each of the back-end servers if the ACLs exist on the top-level object of the partition split point.

About this task

Security Verify Access must have proper access control to allow it to manage users and groups within the suffixes where user and group definitions are maintained. To set the necessary ACLs on the back-end servers to allow Security Verify Access to manage the partition suffixes, use the Security Verify Access **ivrgy_tool** utility with the `add-acls` parameter.

Procedure

1. Run the **ivrgy_tool** utility from any system where the Security Verify Access Runtime component is installed.
   For example, the system where the policy server is installed.
2. To apply the proper ACLs on each of the back-end servers, run the following command:

```
ivrgy_tool -h backend_host -p backend_port -D ldap_admin_DN \
-w ldap_admin_pwd -d [-Z] [-K ssl_keyfile] [-P ssl_keyfile_pwd] \
[-N label] add-acls domain
```

For more information about the **ivrgy_tool** utility, see the Reference topics in the IBM Knowledge Center.

Results

The policy server is the only Security Verify Access component that must be retargeted to the Security Directory Server proxy server as described in "Security Verify Access configuration with the proxy" on page 42. Other Security Verify Access components, such as the authorization server or WebSEAL, do not need to be retargeted.

After the policy server is configured, other Security Verify Access components can be configured normally.

When you configure Security Verify Access Runtime for other components, the Security Directory Server proxy server host name and port must be specified for the LDAP host name. It is not necessary to indicate any of the back-end servers.

Chapter 5. Security Verify Access registry adapter for WebSphere federated repositories

The Security Verify Access registry adapter for WebSphere federated repositories uses the Security Verify Access Registry Direct Java API to perform registry-related operations.

The adapter:

- Is a virtual member manager (VMM) adapter. For detailed information about VMM, see the Virtual member manager documentation in the IBMWebSphere Application Server IBM Knowledge Center .
- Supports a single Security Verify Access domain. However, the Security Verify Access supports multiple secure domains support when configured with the LDAP registry.
- Supports the Security Verify Access registries supported by the Registry Direct Java API.

Index

U