IBM @server pSeries AIX

Configuring an AIX Client System for User Authentication and Management Through LDAP

Yantian Tom Lu, Ph.D.
IBM Corporation
11511 Burnet Road
Austin, TX 78758

March 24, 2003

Configuring an AIX Client System for User Authentication and Management Through LDAP

1. Overview

This paper focuses on configuring AIX® systems as clients of directory servers, both IBM Directory servers and third party LDAP (Lightweight Directory Access Protocol) servers, to provide readers with a complete picture of how to configure and exploit the AIX LDAP security solution. A separate paper discusses how to configure the IBM Directory server for user authentication in AIX[1].

AIX first implemented a LDAP security load module in version 4.3[2]. The implementation worked well in a uniform AIX environment. However, users have found it hard to configure AIX systems to work with third party LDAP servers. This shortcoming is primarily the result of the proprietary schema used by AIX[1].

Since AIX 5L™ version 5.2, AIX supports the schema defined in RFC 2307 which is widely used among IBM peers and which is becoming the industry standard for network entities. The schema defines attributes and object classes for such entities as users, groups, networks, services, hosts, protocols, rpc, etc[3]. The RFC 2307 schema is often referred to as the nisSchema. Both of these terms are used interchangeably in this paper.

Client support for the nisSchema in AIX is part of Configurable Schema Support Mechanism (CSSM), which is a bigger effort to support arbitrary schema. With CSSM, AIX systems can be configured to support LDAP directory servers using any schema. At present, CSSM is implemented for users and groups only.

Configuring AIX to do naming lookup through LDAP for network entities, including users and groups, is also implemented in AIX 5L v5.2. However, this paper deals only with issues related to user authentication and user/group management through LDAP. Naming lookup services for other network entities is addressed in a separate paper[4].

This paper addresses only client configuration. Section 2 introduces the major components and their functionality in an AIX LDAPclient system. Section 3 gives step-by-step instruction on configuring an AIX client system. In Section 4, detailed behaviors and new features of the AIX LDAP client, including CSSM are presented and discussed. System management in respect of the LDAP load module and detailed steps to enable LDAP user authentication are given in Section 5.

2. The Components

2.1 ldap.client

This is the fileset for IBM Directory client. This fileset has to be installed for LDAP authentication to work. The ldap.client fileset is shipped on the AIX 5L v5.2 base CDs.

2.2 /usr/lib/security/LDAP

This is the LDAP load module. The 64-bit version of it is /usr/lib/security/LDAP64. When a library call is routed to LDAP, the LDAP load module is loaded. Both modules are installed by default as part of base operating system.

2.3 /usr/sbin/mksecldap

This is the script to configure an AIX client system. It is owned by user root and group security, with permission set to 500. This script can also be used to configure an IBM Directory server on an AIX system.

2.4 /usr/lib/security/methods.cfg

This is the file where the loadable authentication module is defined. **mksecldap** adds the following stanza to enable the LDAP loadable module during client setup:

```
LDAP:
      program = /usr/lib/security/LDAP
      program_64 = /usr/lib/security/LDAP64
```

2.5 /etc/security/ldap/ldap.cfg

This is the configuration file for **secldapclntd** client daemon. It contains information for **secldapclntd** to function correctly, e.g., host names of LDAP servers, bindDN and password, server port, SSL key location, etc. The file is updated by **mksecldap** command during client setup. The default ldap.cfg file shipped with AIX can be found at Section 8.1.

2.6 /usr/sbin/secldapclntd

This is the client daemon that communicates with the LDAP server. It is owned by user root and group security, with permission set to 500. It can only be started by the root user or a privileged process like init. **mksecldap** starts this daemon after a successful client setup.

3. Client Configuration

3.1 The mksecldap command

mksecldap is an AIX command for IBM Directory server and client setup. The syntax for client setup is:

```
mksecldap -c -h server -a bindDN -p bindpwd [ -d baseDN ] [ -n serverport ] [ -k
SSLkeypath ] [ -w SSLkeypasswd ] [ -t cacheTTL ] [ -C cachesize ] [ -P
numberOfThread ] [ -T heartBeatInt ] [ -u userlist ] [ -U ]
```

The **mksecldap** command performs the following tasks for client configuration:

Saves the LDAP server(s)' host name

Saves the baseDNs for users and groups.

If the server contains network information data, saves the baseDNs for hosts, networks, services, protocols, netgroup, and rpc.

If the network information data is found to exist from the LDAP server, updates the */etc/netsvc.conf*, */etc/irs.conf*, and */etc/rpc.conf* files with the nis_ldap resolver for naming resolution through LDAP.

Sets communication to SSL if the -k and -w options are supplied.

Saves the admin DN and password.

Sets the cache size limit, if supplied from command line.

Sets the cache TTL value, if supplied from command line.

Sets the number of threads used by the **secldapclntd** daemon, if supplied from command line.

Sets the list of users to use LDAP by changing value of the **SYSTEM** and **registry** attributes to LDAP and saves the value to the */etc/security/user* file, if -u option is supplied from command line.

Updates the */usr/lib/security/methods.cfg* file with the following LDAP stanza:

```
LDAP:
      program = /usr/lib/security/LDAP
      program_64 = /usr/lib/security/LDAP64
```

Starts the **secldapclntd** client daemon.

Adds an entry to the */etc/inittab* file so that the **secldapclntd** daemon will be started after reboot.

Steps 3 and 4 are new to AIX 5L v5.2 and later only.

Flags:

-a adminDN	Specifies the LDAP server administrator DN. It must match the one used for the server setup.
-c	Indicates the command is being run to setup a client.
-C cachesize	Specifies the maximum number of user entries used in the client side daemon cache. Valid values are 100-10,000 for user cache. The default value is 1,000. The group cache size is 10% of that of user cache.
-d baseDN	Specifies the suffix or base DN for the **mksecldap** command to search for base DNs for users, groups, and other network information entities. If not specified from the command line, the entire database is searched.
-h serverlist	Specifies a comma separated list of LDAP server host names (server and backup servers).
-k SSLkeypath	Specifies the full path to the SSL key database.

-n serverport	Specifies the port number that the LDAP server listens to.
-p adminpasswd	Specifies the clear text password for the administrator DN of the LDAP server. It must match the one used for the server setup.
-P numberofTreads	Specifies the number of threads the client side daemon uses. Valid values are 1-1,000. The default is 10.
-t cacheTTL	Specifies the maximum time length that a cache entry expires. Valid values are 60-3,600 seconds. The default is 300 seconds. Set this value to 0 to disable caching.
-T heartBeatInt	Specifies the time interval of heartbeat between this client and the LDAP server. Valid values are 60-3,600 seconds. Default is 300.
-u userlist	Specifies the comma separated list of usernames. Specify ALL to enable all users on the client.
-w SSLkeypasswd	Specifies the password for the SSL key.
-U	Specifies to undo the previous server setup to the /etc/security/ldap/ldap.cfg configuration file.

All setup information is saved to the */etc/security/ldap/ldap.cfg* file.

3.2 Client Setup Examples

Here are a few examples:

Example 1:

```
# mksecldap -c -h monster -a cn=admin -p adminpwd
```

This sets up the local host as a client of the LDAP server running on host monster. `cn=admin` and `adminpwd` are the LDAP server administrator DN and password respectively.

Example 2:

```
# mksecldap -c -h sky.ibm.com -a cn=admin -p adminpwd -n 2023
```

This sets up the local host as a client of the LDAP server running on host sky.ibm.com. The `-n 2023` indicates that the server is listening on port 2023.

Example 3:

```
# mksecldap -c -h monster,sky.ibm.com -a cn=admin -p adminpwd -d o=mycompany
```

This sets up the local host as a client of the LDAP servers running on host monster and host sky.ibm.com. The -d option instructs **mksecldap** to search for user and group data under the `o=mycompany` base DN (domain). When multiple domains exist, the -d option has to be used so that **mksecldap** configures the client against a specific subtree domain. Without the the -d option, **mksecldap** uses the first domain that it finds.

Example 4:

```
# mksecldap -c -h monster -a cn=admin -p adminpwd -k /usr/ldap/key.kdb -w
keypwd
```

This sets up the local host as a client of the LDAP server running on host monster using SSL secure communication. The SSL key is `/usr/ldap/key.kdb` and the password to the key is `keypwd`. To run this command, one has to install the GSKit package, and create the client key using the key generation tool from the package. For more information on installing GSKit and generating the SSL key, see the **Secure Communication** section.

Example 5:

```
# mksecldap -c -h monster -a cn=admin -p adminpwd -u user1,user2
```

This sets up the client and enables user1 and user2 to login using LDAP. **mksecldap** sets **SYSTEM** and **registry** attributes to LDAP for user1 and user2. The **SYSTEM** attribute controls the login authentication path, and the **registry** attribute indicates where a user is administered. If **-u ALL** is used in place of **-u user1,user2**, then all locally defined users will have their **SYSTEM** and **registry** attributes set to LDAP. These users have to be defined in the LDAP server to login to the local system, otherwise, they will get the following message when they try to login:

3004-007 You entered an invalid login name or password.

Examples 1 - 4 set up the local host to be a client of a LDAP server(s), but do not enable LDAP authentication mechanism for users. To enable LDAP login, either use the -u option shown in example 5 above or refer to the **LDAP User Management** section later in this paper.

3.3 Manual Configuration

3.3.1 Configuring secldapclntd Daemon

Starting from AIX 5L v5.2, a default */etc/security/ldap/ldap.cfg* client configuration file is shipped, with all the entries in the file commented out. The default configuration file has all the necessary entries and description for each entry to facilitate manual configuration process. The minimum set of attributes that has to be set:

Attributes	Description
ldapservers	Comma separated list of LDAP servers this client will talk to
ldapadmin	LDAP server administrator DN or privileged bindDN
ldapadmpwd	Administrator bind password
userattrmappath	Path to user attribute map
groupattrmappath	Path to group attribute map
idattrmappath	Path to id attribute map
userbasedn	Parent DN where user entries are stored
groupbasedn	Parent DN where group entries are stored

idbasedn	Parent DN where id entry is stored
ldapport	Port that LDAP server listens to

3.3.2 Enabling LDAP Loadmodule

Edit the */usr/lib/security/methods.cfg* file, and append the following stanza to it:

```
LDAP:
      program = /usr/lib/security/LDAP
      program_64 = /usr/lib/security/LDAP64
```

3.3.3 Start secldapclntd Daemon

Simply enter **secldapclntd** at command line to start the daemon:

```
# /usr/sbin/secldapclntd
```

Once **secldapclntd** daemon is running, try to test the configuration by running the **lsuser** command:

```
# /usr/sbin/lsuser -R LDAP uname
```

Where *uname* is the user name of an account defined in LDAP. If the user is listed, the configuration is successful. Otherwise, if you are sure uname exists in LDAP and lsuser fails to list the user, configuration is not correctly done. Revisit the */etc/security/ldap/ldap.cfg* file and make corrections to any misconfiguration, restart the **secldapclntd** daemon and try the command again.

3.3.4 Update */etc/inittab*

Run the following command to add **secldapclntd** daemon to */etc/inittab* file. This makes the init process start **secldapclntd** daemon automatically after reboot.

```
# /usr/sbin/mkitab "ldapclntd:2:once: /usr/sbin/secldapclntd >dev/console 2>&1"
```

4. System Behavior

4.1 Multiple LDAP Server Support and Failover Mechanism

Multiple servers can be supplied to the **mksecldap** command. Those servers can be master and its replicas or peer-to-peer servers. It is required that all of the servers share the same administrator DN and password. During startup, the **secldapclntd** client daemon makes connection to all of the servers. Usually, secldapclntd sets the first server it successfully connects to as the current server - the server that it will actually communicate to. Whenever the current server becomes unavailable, **secldapclntd** automatically turns to the next active server. This failover is immediate and is transparent to users.

This failover mechanism does require that all servers must use the same bindDN and bind password, and same SSL key certificate and key password if SSL is configured.

The **secldapclntd** daemon also keeps checking each server status by contacting the server periodically. The default period is every 300 seconds. This value can be reset by modifying the heartbeat interval value in the */etc/security/ldap/ldap.cfg* file. Valid values are 60 seconds to an hour.

Once the **secldapclntd** daemon detects that a failed server is back on line, it binds to the server again for later use. However, it will not make this server the current server. Instead, it continues to use the server it was talking to.

4.2 Client behavior when the Master Server is Down

In a master replica directory setup, all add/delete/update operations have to be done on the master server, and the changes are then propagated to replicas. One question users often ask is what happens if the master is down, can users still login to client systems? The answer is yes, users are still able to login to client systems if any replica is still active.

The AIX login process needs retrieval from as well as update to the the LDAP server. The update is done by the login process to save login information, e.g., tty from which last login is attempted, success or failure of the login, failed login count, etc. However, there is no login hard fail due to potential failure of such update, and as a result login can still proceed when the master server is off-line. The only concern is the loss of such login attempt information.

However, this may not be a concern for you if your server does not support these attributes (e.g., RFC2307 schema type). In this case, login attempt information is never written to the directory server.

4.3 Multiple Domain Support

An AIX system can be configured as a client to a specific domain. By simply modifying the baseDNs where this client gets data from, one can reconfigure the system to be a client to a different domain. One can do this by simply running mksecldap with the "-d *newbasedn*" option to reconfigure the client, where *newbasedn* is the parent DN which contains users and groups of the new domain. See example 3 of client setup in section 3.2.

4.4 nisSchema (RFC2307) Schema Support

In addition to the continuing support for LDAP servers with AIX user specific schema, AIX also added support for schema defined in RFC 2307. At client configuration time, **mksecldap** command will detect the schema types used by the LDAP server. **mksecldap** does this by querying the LDAP server for posixAccount, aixAccount, and aixAuxAccount objectclasses. The following table shows the schema types and related objectclasses and attribute maps.

Schema Type	Objectclass	Maps
RFC2307	posixAccount	2307user.map, 2307group.map
RFC2307AIX	posixAccount + aixAuxAccount	2307aixuser.map, 2307aixgroup.map, aixid.map

AIX	aixAccount	aixuser.map, aixgroup.map.

Once the schema type is determined, **mksecldap** sets up the corresponding maps that will be used by the LDAP load module to correctly map AIX attributes to LDAP attributes and vise versa. AIX ships three sets of maps as shown in the above table.

With the new support for nisSchema, AIX client can be configured to any LDAP server which is RFC 2307compliant. Configuring an AIX system to talk to a third party LDAP server is the same as configuring it to talk to an IBM Directory server.

4.5 Configurable Schema Support Mechanism (CSSM)

AIX extended its schema support by implementing a configurable schema mechanism. Actually, support for each specific schema, including the new support for nisSchema, is based on CSSM. This mechanism makes AIX systems independent of any hard coded schema.

CSSM is implemented through a set of maps - a user map, a group map, and an id map. The user map defines the AIX - LDAP attribute mapping for user attributes; the group map defines the AIX - LDAP attribute mapping for group attributes; and the id map defines the AIX - LDAP attribute mapping for user IDs and group IDs.

Of the three maps, the meaning for user map and group map is obvious. The id map contains the schema for user and group ids to be assigned to the next new user and group. RFC 2307 schema does not support id entity and therefore does not require the id mapping.

A map file has multiple entries, each of which maps an AIX attribute to a corresponding LDAP attribute. An entry contains the following space separated fields:

```
AIX_attribute_name AIX_attribute_type LDAP_attribute_name LDAP_attribute_type
```

- `AIX_attribute_name`: specifies the attribute name used by the AIX operating system. See the */usr/include/usersec.h* for attribute names and types.
- `AIX_attribute_type`: specifies the attribute type of the AIX attribute. Valid values are SEC_CHAR for string type, SEC_INT for integer type, SEC_LIST for list type, and SEC_BOOL for boolean type.
- `LDAP_attribute_name`: specifies the LDAP attribute name which represents the AIX_attribute_name
- `LDAP_attribute_type`: value type of the LDAP attribute. Valid values are "s" for single-valued and "m" for multi-valued.

AIX ships three set of maps to support three schema types - RFC2307, RFC2307AIX, and AIX. (See above table). **mksecldap** command is able to detect the schema types used by a LDAP server and set the client to use the correct set of maps. See Section 8.2 for map examples of RFC2307AIX schema type.

Manual steps have to be taken to configure a client system to work with a LDAP server which uses schema different from the three known types. These steps involve creation of new maps, making the client daemon use the newly created maps, and steps for more general configuration. Here are the steps to construct new maps:

Make a copy of the existing AIX maps, aixuser.map, aixgroup.map, and aixid.map, in the */etc/security/ldap* directory. e.g., name the new copies myuser.map, mygroup.map, and myid.map.

Edit each of the map copies, remove any entries whose AIX attribute is not supported by your LDAP server. However, there is a minimum set of required attributes which has to be supported by the LDAP server.

For each entry in a map file, make necessary changes to the LDAP_attribute_name according to your server schema, e.g., if the AIX **username** is to be represented by LDAP **cn**, one would change the username entry:

From:
```
username      SEC_CHAR    username     s
```

to:
```
username      SEC_CHAR    cn           s
```

Also make necessary changes to the LDAP_attribute_type, set it to "s" if it is single-valued, "m" for multi-valued.

Once changes are made to all the new maps, edit the */etc/security/ldap/ldap.cfg* client daemon configuration file to use these new maps. In our example, we would change the following entries:

```
userattrmappath: /etc/security/ldap/aixuser.map
groupattrmappath: /etc/security/ldap/aixgroup.map
idattrmappath: /etc/security/ldap/aixid.map
```

to:
```
userattrmappath: /etc/security/ldap/myuser.map
groupattrmappath: /etc/security/ldap/mygroup.map
idattrmappath: /etc/security/ldap/myid.map
```

4.6 Client Caching

To reduce network traffic and improve performance for both servers and clients, AIX implements a client cache for users and groups. When the **secldapclntd** daemon receives a request, it searches its cache for the user or group entry first, if found the request is fulfilled using the cached values; if not found, **secldapclntd** makes a LDAP call to the server. Once it receives the reply from the server, the **secldapclntd** daemon updates its cache with the entry and returns the entry to the caller.

By default, the **secldapclntd** daemon maintains cache for maximum of 1000 users and 100 groups.

However, these values can be overwritten by modifying the **usercachesize** and **groupcachesize** in the */etc/security/ldap/ldap.cfg* file. Valid values are 100 - 10,000 entries for users and 10 - 1,000 for groups.

The cache TTL value can be set by modifying the **cachetimeout** attribute in the *ldap.cfg* file. Valid values are 60 - 3,600 seconds. Default is 300 seconds. If this value is set to 0, the caching mechanism is disabled.

4.7 Multi-threading

The **secldapclntd** is a multithreaded program. The number of threads used is configurable by changing the **numberofthread** variable in the *ldap.cfg* file. Valid values are 1 - 1,000; default is 10. Under most cases, using the default is recommended. This configuration is left open for experienced administrators to fine tune the system performance. Many factors can affect one's choice, e.g., the hardware, and the workload. Too many threads may result in heavier overhead and may actually decrease the performance.

4.8 Failed Login Count

All account information is saved to the LDAP server, regardless of which client login is attempted or user management is done. For example, if a password change is done from a client, the new password is saved to the server and is visible to all other client systems. Once a LDAP user's account is locked, no login of the user is permitted from any client system. Furthermore, failed login count is accumulative from any client system. For example, if an administrator sets the **logintries** attribute of an LDAP account after 5 failed login attempts, and someone attempted three times from client A, and then moved to client B and tried 2 more attempts, but all attempts are failed, the failed login count will be set to 5 and user is denied login to any of the client systems.

This behavior is persistent as long as your LDAP server supports these attributes. If one uses IBM Directory server and configures the server to use the **RFC2307AIX** schema type (RFC 2307 with full AIX schema support) or **AIX** schema type, one will get this behavior. For servers using pure **RFC2307** schema or schema which does not support these attributes, one won't get this behavior.

4.9 Account Locking

An account can be locked by settting its **account_locked** attribute to true. One thing to mention here is that the AIX **account_locked** attribute is mapped to the LDAP **isaccountenabled** attribute directly (i.e., account_locked=true is mapped to isaccountenabled=true). The mapping between the two is a direct mapping as shown in the following table:

ACTION	AIX	LDAP
lock an account	account_lock = true	isaccountenabled = true
unlock an account	account_lock = false	isaccountenabled = false

The command

```
# chuser -R LDAP account_locked=true foo
```

will lock foo's account by setting **isaccountenabled** to true. It is highly recommended that one administers users using the AIX commands and ignore the data structure stored in the LDAP server. However, if one does need to manipulate user data directly to the LDAP server through, for example, the LDAP interface or LDAP commands, keep in mind to set **isaccountenabled** to true to lock the user account.

5. LDAP-User Management

5.1 The SYSTEM Attribute

After configuring an AIX system using **mksecldap**, one more step is needed to enable user login through LDAP. As already discussed earlier, user authentication is controlled by the **SYSTEM** variable of the */etc/security/user* file. The AIX administrator has to set a LDAP user's **SYSTEM** attribute to LDAP for the user to login to the local system. For example, to enable LDAP user foo to login to the system, run:

```
# chuser SYSTEM=LDAP registry=LDAP foo
```

To allow all LDAP users to login to the system, one can set each LDAP user's **SYSTEM** attribute to LDAP, or simply set the default stanza's **SYSTEM** attribute to LDAP (one has be do this manually to the */etc/security/user* file). However, the implication of the two is different. By setting each user's **SYSTEM** attribute to LDAP, an administrator explicitly sets that these users to login through LDAP, with the implication that any newly defined local users whose **SYSTEM** is not set can continue to login through **compat** mechanism (or whatever the default **SYSTEM** attribute is set to). By setting the default **SYSTEM** to LDAP, an administrator implicitly defines that all users whose **SYSTEM** is not set to any value will login through LDAP.

At the login time, the login process checks for the user's **SYSTEM** attribute to authenticate the user accordingly. If a user's SYSTEM is not defined, the default **SYSTEM** value is used.

5.2 The registry Attribute

AIX user management is controlled by the **registry** attribute, which specifies where the user is administered. If a user account exists both locally and in LDAP, AIX requires both the **SYSTEM** and **registry** be set to the same value to work correctly. The only exception to this rule is that each LDAP account is unique, i.e., no corresponding local account exists.

The **registry** determines which user database to use when managing users and it also determines the location where login activity is logged. By setting the **SYSTEM** and **registry** differently, one will run into the risk that the user is defined to login through one mechanism, but the user's account activity is logged to a different user database. It may also result in root managing the wrong account when changing password, locking the account, etc.

A safe way of avoiding such problems is to set the **SYSTEM** and **registry** to the same value, e.g.,

```
# chuser SYSTEM=LDAP registry=LDAP foo
```

This is especially important for the default stanza of the */etc/security/user* file. One has to remember to add "registry = files" to the root stanza if the default **registry** is set to LDAP, otherwise root may not be able to login to the system.

Although multiple authentication mechanisms can be configured (through the **SYSTEM** and **registry** attributes), it is highly recommended that each system only enable one mechanism to simplify user management and minimize related security issues (see section 5.3).

If one does need both local and LDAP login, do the following to minimize any side effect:

Synchronize locally defined users with LDAP users, or

Create unique LDAP users with no corresponding local users - no sharing of same id between local and LDAP users.

5.3 Security

To prevent unauthorized access to resources that one should not normally be allowed to, it is highly recommended that a system be configured to use a single authentication mechanism. Currently there are few tools to synchronize users/groups across multiple user registries. Under most cases users and groups are out of sync between registries. Allowing multiple authentication mechanisms can result in a user having unauthorized access to resources.

Such unauthorized access to resources is due to user id "conflict" between two user registries - two user accounts defined in two different registries but they "share" the same numeric user id. When a user login to one of the accounts, he will be able to access the files that are really owned by another user defined in a different registry. An administrator has to take steps to avoid such conflict when enabling a system for multiple authentication mechanisms.

5.4 Host Login Control

As described in Section 5.3, an easy way to enable LDAP authentication is to set the default **SYSTEM** attribute to LDAP. However, this potentially enables login of all users defined remotely in LDAP. To prevent such from happening, AIX has enforced a host login control mechanism.

Since AIX 5L v5.2, two new attributes, **hostsallowedlogin** and **hostsdeniedlogin**, are introduced for this purpose. These two new attributes are user attributes and therefore host login control can be defined on a per user bases. Existing high level commands, e.g., **mkuser**, **chuser**, and **lsuser**, support these two new attributes.

Here are some examples of setting which hosts a user can or cannot login to:

```
# mkuser -R LDAP hostsallowedlogin=monster foo
```

This creates a new user account foo, and sets that user foo is allowed to login only to host monster.

```
# chuser -R LDAP hostsdeniedlogin=monster foo
```

This sets that user foo is denied login to host monster, but is allowed to login to any other hosts.

```
# chuser -R LDAP hostsallowedlogin=124.212.23/24 foo
```

This sets that user foo is allowed login to any hosts in the network 124.212.23./24.

A few other notes when using the host login control mechanism:

hostsallowedlogin and **hostsdeniedlogin** can be used jointly.

In case of conflict, **hostsdeniedlogin** takes precedence.

If neither is defined, the default values will be checked and login decision made.

If neither the user's or the defaults get set, user is allowed to login to any system. This is for backward compatibility.

To set default behavior for host login control in LDAP, find the default entry from the LDAP user database, and add the **hostsallowedlogin** and/or **hostsdeniedlogin** attributes with appropriate values. This can be done through DMT or by running the **ldapmodify** command.

It is important to understand that such host login control is only effective for AIX 5L v5.2 or later. AIX 5L v5.1 and AIX v4.3 systems are not aware of the **hostsallowedlogin** and **hostsdeniedlogin** attributes and are not controlled by this mechanism. This is another reason for upgrade your AIX 5L v5.1 and AIX v4.3 systems to AIX 5L v5.2.

5.5 Default Entry in LDAP

Each LDAP user database has a default entry, just as the local default stanza of */etc/security/user*. The LDAP default entry is stored together with all other user entries under the same user base DN. The default entry's RDN is "uid=default" if the server is RFC 2307 compliant, or "username=default" if the server uses AIX specific schema (see Server Configuration paper). The default entry is for internal use only, no one is allowed to login or bind using that entry.

The function of the default entry is exactly the same as the local default stanza of the */etc/security/user* file, with the default entry being used only in the LDAP environment and the local default stanza being used in compat (local+NIS) mode. The only difference is that the **SYSTEM** and **registry** attributes are always stored/retrieved from the local default stanza. These two attributes determine the local system authentication and user management behavior, and do not mean anything if stored in the LDAP database.

Unlike a regular LDAP user account, the default entry is not treated as a user account by AIX, and AIX high level commands do not work on the default entry. To add/delete/set an attribute of the

default entry, one has to use the DMT or the **ldapmodify** command.

6. Daemon Commands

The **secldapclntd** client daemon is an important part of the AIX LDAP security subsystem. The following commands are provide for managing this daemon.

start-secldapclntd - starts the **secldapclntd**, this is the same as entering secldapclntd directly from command line.

stop-secldapclntd - stops the **secldapclntd** daemon

restart-secldapclntd - stops the current running **secldapclntd** daemon, and restarts it. If **secldapclntd** is not running, this works the same as **start-secldapclntd**.

ls-secldapclntd - lists the **secldapclntd** daemon statues, including current server that is is talking to, port number, caching status, etc.

flush-secldapclntd - clears the cache of the **secldapclntd** daemon.

7. Secure Communications

IBM Directory supports secure communication between server and client through Secure Sockets Layer (SSL). To use SSL, one needs to install the ldap.max_crypto_client fileset and the GSKit package[5] to each client system from the AIX expansion pack.

Follow these steps to generate the client key database, import the server certificate, and configure a client to use SSL:

Install GSKit on each client system.

Run **gsk5ikm** to generate the key database on each client (see Server Configuration paper[1] for details on key generation).

Copy the server certificate to each of the clients.

On each client system, run **gsk5ikm** to import the server certificate to the key database:

a. Enter **gsk5ikm** to start the tool

b. Click **Key Database File** menu bar and select **Open**.

c. Enter the path and name of the existing key database file then click **OK**.

d. Enter the password.

e. Ensure **Signer Certificates** is chosen and Click on **Add**.

f. Enter the name and location of the server's certificate file

g. Enter a label for the server certificate entry in the client's key database file, for example, 'Corporate Directory Server' then click **OK**.

h. click **View/Edit**, mark **Set the certificate as a trusted root**, then click **OK**.

Enable SSL for the LDAP server (see the Server Configuration paper[1]).

Enable SSL for each client:

```
# mksecldap -c -h servername -a adminDN -p pwd -k /usr/ldap/etc/mykey.kdb -p keypwd
```

where `/usr/ldap/etc/mykey.kdb` is full path to the key database, `keypwd` is the password to the key. If key password is not entered from command line, a stashed password file from the same directory will be used. The stashed file needs to have the same name as the keydatabase with an extension of .sth (e.g., mykey.sth).

In the above step 4, if the server SSL uses a self-signed certificate, the certificate needs to be exported and taken to each client. Following are the steps to export the self-signed certificate from the server:

Enter **gsk5ikm** to start the tool

Click **Key Database File** menu bar and select **Open**.

Enter the path and name of the existing key database file; then click **OK**.

Enter the password.

Ensure **Personal Certificates** is chosen and Click on **New Self-Signed Certificate**.

Enter **Key Label**, **Common Name**, **Organization**, select **Version** and **Key Size**; then click **OK**.

Select the new certificate you just created, click **Extract Certificate**.

Select the **Data Type**, enter the Certificate **file name** and Location; then click **OK**.

8. Appendix

8.1 /etc/security/ldap/ldap.cfg

```
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos520 src/bos/etc/security/ldap/ldap.cfg 1.5
#
# Licensed Materials - Property of IBM
#
# Restricted Materials of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 2002
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# secldapclntd LDAP client daemon configuration file

# Comma separated list of ldap servers this client talks to
#ldapservers:myldapserver.ibm.com

# LDAP server bindDN
```

```
#ldapadmin:cn=admin

# LDAP server bindDN password
#ldapadmpwd:secret

# Whether to use SSL to communicate with the LDAP server. Valid value
# is either "yes" or "no". Default is "no".
# Note: you need a SSL key and a password to the key to enable this.
#useSSL: no

# SSL key file path and key password
#ldapsslkeyf:/tmp/key.kdb
#ldapsslkeypwd:mykeypwd

# AIX-LDAP attribute map path.
#userattrmappath:/etc/security/ldap/aixuser.map
#groupattrmappath:/etc/security/ldap/aixgroup.map
#idattrmappath:/etc/security/ldap/aixid.map

# Base DN where the user and group data are stored in the LDAP server.
# e.g., if user foo's DN is: username=foo,ou=aixuser,cn=aixsecdb
# then the user base DN is: ou=aixuser,cn=aixsecdb
#userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata
#groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata
#idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata
#hostbasedn:ou=hosts,cn=nisdata,cn=aixdata
#servicebasedn:ou=services,cn=nisdata,cn=aixdata
#protocolbasedn:ou=protocols,cn=nisdata,cn=aixdata
#networkbasedn:ou=networks,cn=nisdata,cn=aixdata
#rpcbasedn:ou=rpc,cn=nisdata,cn=aixdata

# LDAP class definitions.
#userclasses:aixaccount,ibm-securityidentities
#groupclasses:aixaccessgroup

# LDAP server version. Valid values are 2 and 3. Default is 3.
#ldapversion:3

# LDAP server port. Default to 389 for non-SSL connection and
# 636 for SSL connection
#ldapport:389
#ldapsslport:636

# Follow aliases. Valid values are NEVER, SEARCHING, FINDING, and
# ALWAYS. Default is NEVER.
#followaliase:NEVER

# Number of user cache entries. Valid value is 100 - 10000 entries.
# Default is 1000.
#usercachesize: 1000

# Number of group cache entries. Valid value is 10 - 1000 entries.
# Default is 100.
#groupcachesize: 100

# Cache timeout value in seconds. Valid value is 60 - 60*60 seconds.
# Default is 300. Set to 0 to disable caching
```

```
#cachetimeout: 300

# Time interval in seconds that the secldapclntd daemon contact the
# LDAP server for server status. Valid value is 60 - 60*60 seconds.
# Default is 300.
#heartbeatinterval: 300

# Number of threads the secldapclntd daemon uses to to process jobs.
#Valid value is 1 - 1000. Default is 10
#numberofthread: 10
```

8.2 Attribute Maps

AIX ships 3 set of maps, each for a supported schema type. Here only the maps for RFC2307AIX schema type are listed for reference. Other maps types can be found at the /etc/security/ldap directory in a AIX 5L v5.2 (or later) system.

8.2.1 /etc/security/ldap/2307aixuser.map

```
# ====================================================================
#
# AIX-RFC2307 LDAP user attribute name mapping table
#
# Format:
# AIX_ATTR AIX_ATTR_TYPE LDAP_ATTR LDAP_VALUE
#
# AIX_ATTR: AIX attribute name
# AIX_ATTR_TYPE: AIX attribute type - SEC_CHAR, SEC_INT, SEC_LIST,
#                SEC_BOOL
# LDAP_ATTR: LDAP attribute name
# LDAP_VALUE: LDAP attribute type - "s" for single-valued attribues
#             or "m" for multi-valued attributes.
#
# NOTE:
# In case the client needs to talk to a LDAP server with different
# schema (attributes) mapping than the following, modify the corresponding
# LDAP attributes to match those defined in the new LDAP server, and
# comment out the lines where the attribute(s) is not defined.
#
# Save a copy of this file before you modify entries in this file.
#
# ====================================================================

# The following attributes are required by AIX to be functional
# They are defined in the RFC 2307 schema
# The following attributes are required by AIX to be functional
# They are defined in the RFC 2307 schema
username          SEC_CHAR         uid                     s
spassword         SEC_CHAR         userpassword            s
id                SEC_INT          uidnumber               s
pgrp              SEC_CHAR         gidnumber               s
gecos             SEC_CHAR         gecos                   s
home              SEC_CHAR         homedirectory           s
shell             SEC_CHAR         loginshell              s
lastupdate        SEC_INT          shadowlastchange        s
```

```
# The following attributes are optional
# They are defined in the RFC 2307 schema
maxage            SEC_INT          shadowmax               s
minage            SEC_INT          shadowmin               s
maxexpired        SEC_INT          shadowexpire            s
pwdwarntime       SEC_INT          shadowwarning           s
# The "shadowflag" defined in RFC 2307 is a reserved field.
#reserved         SEC_INT          shadowflag              s
#notused          SEC_INT          shadowinactive          s

# The following attributes are optional
# They are defined in the aixAuxAccount schema
account_locked    SEC_BOOL         isaccountenabled        s
admgroups         SEC_LIST         admingroupnames         s
admin             SEC_BOOL         isadministrator         s
auditclasses      SEC_LIST         auditclasses            s
auth1             SEC_LIST         authmethod1             s
auth2             SEC_LIST         authmethod2             s
auth_domain       SEC_CHAR         altsecurityidentities   s
core              SEC_INT          coresizelimit           s
core_hard         SEC_INT          coresizelimithard       s
cpu               SEC_INT          cpusize                 s
cpu_hard          SEC_INT          cpusizelimithard        s
d_level           SEC_CHAR         aixdefaultmaclevel      s
daemon            SEC_BOOL         isdaemon                s
data              SEC_INT          datasegsize             s
data_hard         SEC_INT          datasegsizehard         s
dce_export        SEC_BOOL         aixisdceexport          s
dictionlist       SEC_LIST         passworddictfiles       s
expires           SEC_CHAR         timeexpirelockout       s
flags             SEC_LIST         passwordflags           s
fsize             SEC_INT          filesizelimit           s
fsize_hard        SEC_INT          filesizelimithard       s
funcmode          SEC_CHAR         aixfuncmode             s
groups            SEC_LIST         grouplist               s
groupsids         SEC_LIST         groupsids               s
herald            SEC_CHAR         herald                  s
herald2           SEC_CHAR         herald2                 s
histexpire        SEC_INT          passwordexpiretime      s
histsize          SEC_INT          passwordhistsize        s
host_last_login   SEC_CHAR         hostlastlogin           s
host_last_unsuccessful_login SEC_CHAR hostlastunsuccessfullogin    s
hostsallowedlogin SEC_LIST         hostsallowedlogin       m
hostsdeniedlogin  SEC_LIST         hostsdeniedlogin        m
l_level           SEC_CHAR         aixlowmaclevel          s
locktime          SEC_INT          locktime                s
login             SEC_BOOL         isloginallowed          s
logindelay        SEC_INT          logindelay              s
logindisable      SEC_INT          logindisable            s
logininterval     SEC_INT          logininterval           s
loginreenable     SEC_INT          loginreenable           s
loginretries      SEC_INT          maxfailedlogins         s
logintimes        SEC_LIST         logintimes              s
logintimes        SEC_LIST         logintimes              s
maxrepeats        SEC_INT          passwordmaxrepeatedchars s
maxulogs          SEC_INT          maxlogin                s
minalpha          SEC_INT          passwordminalphachars   s
```

```
mindiff           SEC_INT          passwordmindiffchars     s
minlen            SEC_INT          passwordminlength        s
minother          SEC_INT          passwordminotherchars    s
nofiles           SEC_INT          openfilelimit            s
nofiles_hard      SEC_INT          openfilelimithard        s
password          SEC_CHAR         passwordchar             s
prompt_mac        SEC_BOOL         aixpromptmac             s
pwdchecks         SEC_LIST         passwordcheckmethods     s
registry          SEC_CHAR         registry                 s
rlogin            SEC_BOOL         isremoteaccessallowed    s
roles             SEC_LIST         rolelist                 s
rss               SEC_INT          physicalmemlimit         s
rss_hard          SEC_INT          physicalmemlimithard     s
sak_enabled       SEC_BOOL         sak_enabled              s
screens           SEC_LIST         aixscreens               s
stack             SEC_INT          stacksizelimit           s
stack_hard        SEC_INT          stacksizelimithard       s
su                SEC_BOOL         isswitchuserallowed      s
sugroups          SEC_LIST         groupswitchuserallowed   s
synonym           SEC_LIST         synonym                  s
sysenv            SEC_LIST         systemenvironment        s
telnet            SEC_BOOL         isremoteaccessallowed    s
time_last_login   SEC_INT          ixtimelastlogin          s
time_last_unsuccessful_login SEC_INT ixtimelastunsuccessfullogin   s
tlogout           SEC_INT          tlogout                  s
tpath             SEC_CHAR         trustedpathstatus        s
tty_last_login    SEC_CHAR         terminallastlogin        s
tty_last_unsuccessful_login SEC_CHAR terminallastunsuccessfullogin s
ttys              SEC_LIST         terminalaccess           s
u_level           SEC_CHAR         aixuppermaclevel         s
uactivity         SEC_INT          timeexpiredlogout        s
umask             SEC_INT          filepermmask             s
unsuccessful_login_count SEC_INT   unsuccessfullogincount   s
unsuccessful_login_times SEC_LIST  unsuccessful_login_times s
usrenv            SEC_LIST         userenvironment          s
utocount          SEC_INT          timeexpirelockout        s
```

8.2.2 /etc/security/ldap/2307aixgroup.map

```
# ==================================================================
#
# AIX-RFC2307 LDAP group attribute name mapping table
#
# Format:
# AIX_ATTR   AIX_ATTR_TYPE   LDAP_ATTR   LDAP_VALUE
#
# AIX_ATTR:      AIX attribute name
# AIX_ATTR_TYPE: AIX attribute type - SEC_CHAR, SEC_INT, SEC_LIST, SEC_BOOL
# LDAP_ATTR:     LDAP attribute name
# LDAP_VALUE:    LDAP attribute type - "s" for single-valued attribues
#                or "m" for multi-valued attributes.
#
# NOTE:
# In case the client needs to talk to a LDAP server with different
# schema (attributes) mapping than the following, modify the corresponding
```

```
# LDAP attributes to match these defined in the new LDAP server, and
# comment out the lines where the attribute(s) is not defined.
#
# Save a copy of this file before you modify entries in this file.
#
# ================================================================================

# The following are required attributes
# They are defined in the RFC 2307 schema
groupname         SEC_CHAR         cn                  s
id                SEC_INT          gidnumber           s
users             SEC_LIST         memberuid           m

# The following are optinal attributes
# They are defined in the aixAuxGroup schema
primary           SEC_LIST         primary             s
adms              SEC_LIST         aixgroupadminlist   s
admin             SEC_BOOL         isadministrator     s
dce_export        SEC_BOOL         aixisdceexport      s
screens           SEC_LIST         aixscreens          s
```

9. References

1. Configuring an IBM Directory Server for User Authentication and Management in AIX- published as a companion white paper.

2. AIX v4.3 Documentation: System Management Guide: Operating System and Devices: LDAP Exploitation of the Security Subsystem.

http://publib.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/baseadmn/toc.htm

3. RFC 2307: An approach for using LDAP as a network information service.

http://www.ietf.org/rfc/rfc2307.txt

4. LDAP naming service in AIX - to be published.

5. Gskit - SSL toolkits home.

http://thetower.tivlab.raleigh.ibm.com/sslhome.htm

6. IBM Directory Server Version 4.1: Administration Guide.

http://www.ibm.com/software/network/directory/library/#4

7. AIX 5L Version 5.2 Security Guide: LDAP exploitation of the Security Subsystem.
http://publib16.boulder.ibm.com/pseries/en_US/aixbman/security/securitytfrm.htm

IBM Corporation
Marketing Communications
Systems Group
Route 100
Somers, New York 10589

Produced in the United States of America
03-03