Apple Certified Specialist - Directory Services
ACS-DS 10.6

NOTEBOOK

Apple Certified Specialist Directory Services 10.6

Disclaimer

This notebook is intended for personal use only, it is a collection of questions and answers regarding the subjects concerning the ACS-DS certification 10.6.

It has never been authorized by Apple or Editors of the Apple Training Series books and is not intended to substitute any official Apple resource suggested to obtain the ACS-DS certification.

The following content is an unofficial (my personal) guide to acquire knowledge to pass the certification exam Snow 301, it may be inaccurate and may contain mistakes, it is just the result of my studies and was intended to help myself in memorizing the main concepts learned reading the official books part of the Apple Training Series.

Marco Massari Calderone

marco at marcomc dot com

Accessing the Local Directory Service

Define the following terms: node, record, attribute

1. Node (domain, directory node, directory domain, directory), directory within a directory service.
2. Record, an individual entry or entity record in a directory; a collection of record attributes and the values or data stored therein.

Identify tools available to view DNS information, including `dig` and `dscacheutil`

Tools using `mDNSResponder` or `DirectoryService`:

1. `ping`
2. `dscacheutil`
3. `dscl`

Tools NOT using `mDNSResponder` or `DirectoryService`:

1. `dig`
2. `nslookup`
3. `host`

List the tools to create and edit local user accounts

1. Workgroup Manager.
2. dsimport:
   ```
   $ dsimport import.txt /Local/default I --template StandardUser
   ```
3. Create a record file (in XML format) in the directory /var/db/dslocal/nodes/Defaults/users.
4. `dscl`:
   ```
   $dscl . -create /Users/newuser
   ```
5. `vipw`, to edit users in /BSD/Local node.

List the master.password and group flat files that hold information for /BSD/Local node

1. /etc/master.password
2. /etc/group

List the location of files that hold information for the /Local/Default node

/var/db/dslocal/nodes/Default

Given a property list file containing a user record, identify the attributes in the record file and their values

1. `gid`
2. `home`
3. `name`
4. `passwd`
5. `realname`
6. `shell`
7. `uid`
8. `generated uid`
9. `smb_id`

Explain the function of the key user attributes in a BSD flat file

1. `name`, user's login name.
2. `password`, user's encrypted password.
3. `uid`, user's id.
4. `gid`, user's login group id.
5. `gecos`, user's full name.
6. `home_dir`, user's home directory.
7. `shell`, user's login shell.

Given a standard installation of Mac OS X 10.6 and Workgroup Manager create a user record in the /Local/Default node

1. From the Server menu choose View Directory.
2. Ensure that you're looking at /Local/Default.
3. Click the Accounts button.
4. Click the New User button.
5. Fill the information needed and click Save.

Given dsimport in a standard installation of Mac OS X 10.6 create a user record in the /Local/Default node

1. Create a text file (import_user.txt) that contains one or more records per user to import with the formatting according to the Standard User Record Template (or a custom template). In case of a custom template you need to specify the template as header of the import file.
2. Run the command:

```
$ dsimport import_user.txt /Local/Default I --template StandardUser
```

Given a standard installation of Mac OS X 10.6, create a user record in the /Local/Default node by copying and modifying a record file

1. Create a new user record in XML format, perhaps using an existing user record as a template.
2. With root privileges copy the user record file to /var/db/dslocal/nodes/Default/users.
3. Copy the shadow password hash file to /var/db/shadow/hash or set up the password and kerberos principal with `sudo password <username>`.

Given `dscl` in a standard installation of Mac OS X 10.6 create a user record in the /BSD/Local node

```
$ dscl . -create /Users/<newuser>
```

List available tools to create and edit local group accounts

1. `dscl`
2. Workgroup Manager
3. Text editors
4. `dseditgroup`

Given an XML file for a group record from the /Local/Default node, identify any group attributes in the records contained in the XML file

1. `name`, start name of the group.
2. `realname`, long name of the group.
3. `gid`, numerical ID to identify the user.
4. `members`, short names of users that are members of the group.
5. `generateduid` (GUID), 128-bit value guaranteed unique across space and time.
6. `smb-id`, SMB primary Group Security ID.
7. `passwd`, usually an asterisk.
8. `groupmemebers`, GUIDs of users that are members of the group.
9. `nestedgroups`, GUIDs of the groups that are members of the group.

Given a BSD group file, identify any group attributes in the file

1. `group`, name of the group.
2. `password`, groups encrypted password.
3. `gid`, the group's decimal ID.
4. `member`, group members.

Identify `dseditgroup` as a tool to create and edit groups in the /Local/Default directory

```
$ dseditgroup -o -create -u <admin name> <new group name>
```

Identify a text editor as a way to create and edit groups in /BSD/Local

Edit the file /etc/group

User dsedit group to modify the groups in the /Local/Default node

Add the user user1 to the group `group1`:

```
$ dseditgroup -o edit -u <admin name> -a user1 -t user group1
```

Identify the location of `DirectoryService` logs

1. /var/log/system.log.
2. /Library/Logs/DirectoryService/DirectoryService.debug.log.

Describe how to enable detailed logging on the `DirectoryService` process

```
$ sudo defaults write /Library/Preferences/DirectoryService/
DirectoryServiceDebug/ “Debug Logging Priority Level” -integer 7
```

Identify `dscl` as a tool to test a `DirectoryService` search path

```
$ dscl /Search -read /Hosts /hostname.domain
```

Identify `kill` and and `killall` as tools to signal a process

1. `$ sudo killall -USR1 <process_name>`
2. `$ sudo kill -USR1 <process_id>`

Enable detailed logging of the `DirectoryService` process

1. `$ sudo killall -USR1 DirectoryService` (toggle logging)
2. `$ sudo touch /Library/Preferences/DirectoryService/.DSLogDebugAtStart` (if troubleshooting issues at startup)
3. `$ sudo touch /Library/Preferences/DirectoryService/.DSLogDebugAtOnce` (if troubleshooting issues at startup)
4. `$ sudo killall -USR2 DirectoryService` (toggle API logging)

Given Terminal display the meaning of a numerical `DirectoryService` error

```
$ dserr <error_id>
```

Identify issues caused by user name collision with the directories

1. The first user record that `DirectoryService` finds in the authentication search path is the record that it uses for authentication, preventing the OpenDirectory Server user to log in successfully.
2. Open Directory attempts authentication ONLY against the first user that it finds int the authentication search path.

Resolve issues caused by user name collision with two directories

Delete the user account from the /Local/Default node

Given Terminal application test the user authentication

```
$ dscl . -authonly <username> <password>
```

Accessing an Open Directory Server

List potential security issues when binding to a directory server

By default the LDAP traffic is unencrypted so all the data that goes across the local network is in clear-text leaving space for a man-in-the-middle attack, to introduce bogus replies to LDAP requests including configuration information.

Define Trusted Binding

Trusted Binding is a mutually authenticated connection between a computer and a directory domain.

Trusted Binding requires Mac OS X 10.4 or later.

List the requirements for Trusted Binding

1. Mac OS X 10.4 or later.
2. The credential of a Open Directory network user with the ability to create a computer record in the shared domain (that is a default capability).

Given Directory Utility, a Mac OS X 10.6 computer and a Open Directory Server, configure the Mac OS X computer to bind to the directory server with trusted binding

1. Click 'Services'.
2. Click edit (authenticate before if necessary).
3. Select the LDAPv3 from the list of services and click the Edit button.
4. Select the server configuration for the Open directory server.
5. Click 'Edit' (enabled only if the server do support binding).
6. Click the 'Bind' button.
7. Enter a computer ID that identifies your computer.
8. Enter the credential of a valid directory user (username and password).
9. Click 'OK' and close the Directory Utility.

Configure the computer bound to an Open Directory server to encrypt all traffic with the directory server

1. Open Directory Utility.
2. Select the server configuration from the LDAPv3 service.
3. Check the box to enable SSL support.
4. Save and close Directory Utility.

Describe the stages in the binding process

1. Enable the LDAPv3 connector.
2. Bind to the Open Directory server.
3. Add the LDAP server to the authentication search path.
4. Join the Kerberos realm.

Describe how Mac OS X auto-configures the search base and schema of Open Directory servers

1. Mac OS X performs a few search to gather information about the object classes supported by the Open Directory server it looks for.
    - `macosxodconfig` (contains configuration information about how to map the Open Directory standard records and attributes to the native records and attributes stored in the LDAP directory.
    - `macosxodpolicy` (policy and security settings).
    - `ldapreplicas` (addresses of the OD master and replicas).
    - Kerberos Client (default realm and the KDC)
2. Using this information gathered from LDAP queries, DirectoryService modifies several files in the directory /Library/Preferences/DirectoryService:
    - `DSLDAPv3PluginConfig.plist` (standard to native records mapping information).
    - `SearchNodeConfig.plist` (directory nodes in the authentication search path).
    - `ContactsNodeConfig.plist` (directory nodes in the contacts search path).
    - `/Library/Preferences/edu.mit.Kerberos` (information about the default realm on the KDC).
    - `/var/db/dslocal/nodes/Default/config/Kerberos:REALM.plist` where REALM is your Kerberos realm).

Identify the client log files that are written to during binding

1. /Library/Logs/DirectoryService/DirectoryService.error.log.
2. /Library/Logs/DirectoryService/DirectoryService.server.log

Define the following terms: search base, authentication search policy and contacts search policy

1. Search base (or search path) is a configurable list of directory nodes that Open Directory must use when searching for a record.
2. Authentication search policy:
   - Authentication is tested only with the first record that Open Directory finds.
   - Once it finds a match in one mode the search is complete and it doesn't consult the next node.
   - If it doesn't find a match in the first node, it performs a search in the second node and so on.
   - Local users always take precedence over network users.
3. Contact search policy:
   - The contacts search path is used only by the applications that are programmed to use it.
   - Open Directory may gather the results of all the directory nodes.

Describe the difference between the search policies, Automatic, Local, Custom

1. Automatic: includes only the LDAP servers assigned by the DHCP server. Not compatible with Trusted Binding. Not compatible with Mac OS X 10.6.
2. Local Directory: use only /Local/Default and /BSD/Local.
3. Custom path: add, reorder and remove directory nodes.

Identify Directory Utility as a tool to configure Mac OS X 10.6 to use the local BSD flat files

1. Open Directory Utility.
2. Select 'BSD Flat File and NIS' from the Services list.
3. Select the "Use User and Group records in BSD local node" checkbox.
4. Click 'OK', then 'Apply' and close Open Directory Utility.

Identify Directory Utility as a method to configure a custom path search policy for directory services

1. Click the 'Search Policy' button.
2. Select the 'Authentication' tab.
3. Choose 'Custom Path' in the Search menu.
4. Add the directory domains you want to add to the search policy.

Identify the client log files that are written during a login attempt

1. /var/log/system.log
2. /var/log/secure.log

Describe the login process for a network user

1. The User must access the `loginwindow` application.
2. The user must provide identification (name) and authentication (password) information.
3. `loginwindow` searches for the user record in the nodes of the authentication path.
4. If the authentication succeed, loginwindow attempts to obtain a TGT from the KDC.
5. If the user authenticate correctly and has authorization to mount the home folder, `loginwindow` mounts the share point hosting the network home folder and sets up the user's graphical environment.

Identify `klist` as a tool for verifying Kerberos authentication

```
$ klist
```

The klist command shows any kerberos tickets that a user possess.

Given `dscl` verify that Mac OS X is able to access user account via the /Search path

```
$ dscl /Search -read /Users/<username>
```

Given the kerberos application verify kerberos authentication

1. Launch the Ticket Viewer application.
2. Click the 'Add Identity' button and provide credential for the Kerberos realm.
3. A successful authentication will display the obtained TGT in list.

Given the Finder verify home directory access

1. Choose 'Go > Connect to Server ...' from the menu.
2. Enter the URL for the home folder.
3. Enter the credential for the network user and choose the share point to mount.
4. The network volume should open in the Finder and the server that is hosting the network volume appears in the Finder's sidebar.

Accessing a Third-Party LDAP Service

Given an OpenLDAP server,create all records necessary for maintaining a network home directory when a user logs into a Mac OS X computer bound to the server

1. Create the attributes required for the mount object:
   - RecordName (ODattribute), (cn, attribute): Server FQDN and full path to shared folder.
   - VFSLinkDir (ODattribute), (native: mountDirectory): path to location where client will mount network volume.
   - VFSType (ODattribute), (native: mountyType):
     - `url` for AFP.
     - `nfs` for NFS.
   - VFSOpts (ODattribute), (native: mountOption):
     - `net` for dynamically mounted shared points.
     - `afp://[user[;AUTH=uname]] <volumename>`.
2. Create a mount object class that contains the mount record attributes.
3. Create a new mount object in your LDAP directory and specify the new attributes.
4. Map the Open Directory standard record type `mount` to the native object class you created in your LDAP directory.

Define the following terms: schema, object class

1. Schema: the rules that determine the object classes and attribute types allowed in your LDAP datastore.
2. Object class (in the schema): define the record types the LDAP directory supports.

List the authentication methods available for 'login window'

1. dsAttrTypeStandard: AuthenticationAuthority
(encrypted password or user's Kerberos and Password Server information).
2. dsAttrTypeStandard: Password
(authentication via password stored in the LDAP server, clear-text or encrypted).
3. LDAP bind if no attribute and use of SSL is recommended:
   - Attempts Kerberos authentication if GSSAPI is supported.
   - Otherwise attempts CRAM-MD5.
   - Otherwise clear-text.

Describe the schema for managed preferences (MCX) records

MCX records have two attributes:

1. MCXFlags (native: apple-mcxflags), when MCX settings exists whether the user can simultaneously login.
2. MCXSettings (native: apple-mcxsettings), XML formatted plist for each category of managed settings.

Identify specific mapping issues with a eDirectory server

1. Make sure that the user you specify to bind with has read access to call the necessary attributes.
   When extending a eDirectory schema you must also update the ACL for the new Object Class and attributes to allow users to access it.
2. LDAP bind password is sent in clear-text.

Identify Directory Utility as a tool to configure client computers to augment LDAP login information with managed client information from Open Directory server.

This configuration is called the "Magic Triangle":

1. You need to configure an Open Directory server with some managed account groups. Make the LDAP directory accounts member of those managed groups.
2. Via Directory Utility bind the Open Directory server to the client (along with the LDAP server) an be sure the OD server is listed in the search paths.
3. The users will still authenticate against the LDAP server but will retrieve managed configurations from the OD server.

Identify Directory Utility as a tool to configure client computers to augment LDAP login information with local static map information

1. Open Directory Utility and authenticate as local administrator (if required).
2. Click 'Services', choose LDAPv3 and click the 'Edit' button.
3. Select the entry for your LDAP service.
4. Click the 'Search & Mapping' tab.
5. Select the attribute you want to add in the left column.
6. Enter the static value (starting with the '#' character) on the right column (use '$' character to include variables inside the static value).

Identify Kerberos command-line tools included in a standard installation of Mac OS X 10.6

1. `klist` command shows any Kerberos tickets that a user possess.
2. `kinit` command is used to obtain a TGT ticket for a specific user.
3. `kdestroy` command destroy any kerberos ticket a user has.

Given a standard installation of Mac OS X 10.6 identify the location of the record and attribute mappings

`/Library/Preferences/DirectoryServices/`

Configure a Mac OS X client computer to supplement LDAP login information with mount information from an Open Directory server

Use the "Magic Triangle" configuration

Describe how to configure a Mac OS X client computer to authenticate to a third-party KDC using Kerberos command-line tools in a standard installation of Mac OS X 10.6

1. Remove the leading four lines of the configuration header from the file `/Library/Preferences/edu.mit.Kerberos` so that Mac OS X does not overwrite it and destroy the changes you will make.
2. Add your KDC to the list of favorite KDC in the `/Library/Preferences/edu.mit.Kerberos` file.

List the Kerberos configuration files in order of precedence

1. `~/Library/Preferences/edu.mit.Kerberos`
2. `/Library/Preferences/edu.mit.Kerberos`
3. `/etc/Krb5.conf`

Describe the effects of using static mappings for an LDAP configuration in Directory Utility

Force Open Directory to use the text you specify as the value, instead of looking up the value from the directory node

Describe the process of user authentication at the login window

1. `loginwindow` identify the user by searching through the authentication search path to find the user record and then it attempts to authenticate the user trying various ways:
   - If the user record contains the AuthenticationAuthority standard attribute (encrypted password, or user's Kerberos and Password Server information) login window uses it.
   - If there is a mapping for the attribute dsAttrTypeStandard: Password, Mac OS X attempts to authenticate with the password that is stored in the LDAP directory,
   - If the record does not have a mapping for either attributes, Mac OS X attempts an LDAP bind to authenticate the user.
2. After a user is successfully authenticated `loginwindow` continues with the login process mounting a share point for the network home folder and checking to see if there are any records for managed preferences.

Describe the process the system uses to apply managed preferences (MCX) policies at login

1. The user is identified and authenticated.
2. Mac OS X queries all directories nodes for group records that the user might belong to and also have managed preferences settings.
3. The OD server returns a list of the user's workgroups to Mac OS X.
4. If the user belongs to more than one group with managed preferences, you must choose one workgroup for the login session.
5. The process that handle the Mac OS X managed preferences combine the managed preferences attributes for the account records along with locally cached managed preferences in those locations:
   - `~/Library/Preferences/com.apple.MCX.plist`
   - `/Library/Managed\ Preferences/<username>`
   - `/Library/Managed\ Preferences/`
   - `/var/db/dslocal/Default/config/mcx.cache.plist`

Describe the process the system uses to locate the user's home folder at login

1. After authentication, select a managed group (if there's a choice).
2. Mac OS X uses the standard attribute `Home Directory` to mount the share point that hosts the network home folder.
3. Mac OS X attempts to mount the share point with the user credentials.
   The OD server that hosts the AFP share attempts to identify the user in its authentication search path. It eventually uses LDAP bind authentication with user's password.
4. The LDAP service allows to bind authentication which tells the AFP server that Mac OS X provided sufficient authentication.
5. The AFP server checks to make sure that the user is authorized to use the AFP service and then allows the Mac OS X computer to mount the share point.
6. The Mac OS X computer mounts the share point in the location specified by `NFSHomeDirectory` attribute (local path for the home folder).

Accessing an Active Directory Service

Define the following terms: computer trust account, domain, forest

1. Domain: it's building block of Active Directory, it is a collection of directory objects such as users, groups and computers.
2. A tree is one or more domains in a contiguous name space.
3. A forest is a set of domain trees that have a common schema and a global catalog.
4. Computer trust account is a special user account given the permissions to bind computers to an AD service. This account is used exclusively to accomplish this function especially if the binding is done by a script (that must include the user password) so that there is not need to expose the password of any user account.

Describe Active Directory packet signing and encryption

SMB packet signing is a security feature designed to prevent man-in-the-middle attacks

Identify Account Preferences, Directory Utility and dsconfigad as tools to bind to an Active Directory server

1. Account Preferences:
   - Click the 'Login Options', click the 'Join' button.
   - Enter the name of the AD domain (not the server name).
   - Fill the information for Active Directory settings.
2. Directory Utility:
   - Select the Active Directory service.
   - Fill the information to the settings dialog, click 'Bind'.
   - It appears the authentication and Computer OU dialog, fill it with a custom container (according to the Organizational Unit the computer is part of).
3. `dsconfigad` command:
   - Bind to AD:
     ```
     $ dsconfigad -a clientname -domain domain.name -u adminname
     -p adpassword -lu localadmin -lp lapasswd
     ```
   - Add the AD to the search path:
     - `$ sudo dscl /Search -create / SearchPolicy CSPSearchPath`
     - `$ sudo dscl /Search -append / CSPSearchPath “Active Directory/All Domains”`
   - Add the AD to the Contacts search path:
     - `$ sudo dscl /Search/Contacts -create / SearchPolicy CSPSearchPath`
     - `$ sudo dscl /Search/Contacts -append / CSPSearchPath “Active Directory/All Domains”`
   - Force DirectoryService to restart and load the new configurations
     ```
     $ sudo killall DirectoryService
     ```

Describe the stages in the binding process to an Active Directory server

1. Mac OS X performs a request for LDAP, Kerberos and Passwd DNS service records in the domain.
2. Mac OS X binds anonymously with LDAP and gather basic Active Directory domain information.
3. DirectoryService's Active Directory connector creates a preliminary Kerberos configuration.
4. Mac OS X uses the Kerberos configuration, authenticates and then requests the nearest domain controller.
5. The domain controller returns a list of the nearest domain controllers, based on the IP subnet of the Mac OS X computer.
6. Mac OS X confirms that it can connect to the LDAP and Kerberos services of the domain controller list.
   DirectoryService and `kerberosautoconfig` create a final kerberos configuration on `/Library/Preferences/edu.mit.Kerberos` and `/var/db/dslocal/nodes/Default/config/Kerberos:REALM.plist`
7. Mac OS X connects to what it was told was the nearest domain controller.
8. Mac OS X searches the domain for an existing computer record and it creates a new computer record to use if it cannot find one.
9. Mac OS X updates its SAMBA machine password and domain SID.
10. Mac OS X updates its DNS record in Active Directory.

List client log files that are written to during binding

`/Library/Logs/DirectoryService/DirectoryService.debug.log` (must be activated).

Given Terminal perform DNS lookups to verify the records required by Active Directory

`$ host _service._protocol.domain.name` (i.e. _ldap._tcp.google.com)

Given `rm` in a standard installation of Mac OS X 10.6 which is to bound an Active Directory, remove all files containing AD bindings informations

1. `$ sudo rm /Library/Preferences/DirectoryService/ActiveDirectory.plist`
2. `$ sudo rm /Library/Preferences/DirectoryService/ActiveDirectoryDomainCache.plist`
3. `$ sudo rm /Library/Preferences/DirectoryService/ActiveDirectoryDomainPolicies.plist`
4. `$ sudo rm /Library/Preferences/DirectoryService/ActiveDirectoryDomainData.plist`

Given a log file, identify any log entries related to an unsuccessful binding

```
$ tail -f /Library/Logs/DirectoryService/DirectoryService.debug.log | grep “Bind Step”
```

Configuring Open Directory Server

Compare and contrast the "Create Users and Groups", "Import Users and Groups" and "Configure Manually" configuration options that are available when you initially configure a Mac OS X Server using Server Assistant

1. Create Users and Groups:
   the server is configured as an Open Directory master when user and group information will be stored and shared with client computers.
2. Import Users and Groups:
   the server is configured as an Open Directory master but is configured to connect to an Open Directory or Active Directory server where user and group records are stored.
3. Configure Manually:
   this option does automatically configure your server to run a DNS service for itself (if you do not specify another DNS service and assign your Mac OS X server IP address) and you will have the ability to bind your server to another directory server or make your server a directory server during initial setup.

List the methods to upgrade a Mac OS X Server

1. Installation optical disc.
2. Netinstall image.
3. Install remotely via Server Assistant.

Given Mac OS X 10.5 Open Directory master, upgrade it to Mac OS X 10.6 without losing any directory information

1. Use Server Admin or `serveradmin` to export service settings to reference.
   Store the exported service settings on a removable drive.
2. Perform an upgrade to Mac OS X Server 10.6.
   After the upgrade the computer restart and Server Assistant leads you through initial setup server setup your existing settings are displayed and you can change them if you like.
3. Enable Kerberos:
   - Use the 'Kerberos' button on the Open Directory pane in Server Admin.
   - Or use `slapconfig -kerberize` command.

     If you have crypt password for the accounts and you don't kerberize them you can use Workgroup Manager to upgrade them.
4. Move the LDAP ACLs because of a change of format the container or record for access controls and ACL information is made available as ReadOnly.
   Use Workgroup Manager to add custom ACLs to the new `olcAccess` attribute (in `olcBDBConfig`).
5. Schema changes must be made under olcSchemaConfig and custom schemas should be added to the `{9}customschema` record. Changes to configure slapd can be made to the back-config backend using inspector in Workgroup Manager on ldap tools. Changes require `slapd` to be restarted.

Identify `changeDirData.pl` as a tool to perform batch updates on an OpenLDAP database

1. The `changeDirData.pl` tool is used to change the contents of the directory data to reflect the new values.
2. It will search the server for the specified value and make the change.
3. It needs to be run in the LDAP Master.
4. The syntax is:

```
$ changeDirData.pl [-hiPv] -s server -u diradmin [-p password] -o oldValue
-n newValue -r recordType
```

Identify `slapconfig` as a tool to promote an OpenDirectory master

```
$ sudo slapconfig -promotereplica diradmin
```

(progress, information is logged to /Library/Logs/slapconfig.log)

Identify Server Admin as a tool to tune OpenDirectory security and performance options.

1. You can use Server Admin to secure LDAP Connection by the creation of a self-signed TLS certificate.
2. In the LDAP tab of the Open Directory service you are given the option to:
   - Enable SSL, via the user of a TLS certificate.
   - Define a Searches timeout to prevent denial-of-service attack.
   - Define the maximum of search results to be returned during a single search to prevent random harvesting of information and to accelerate the speed of every search.

Compare and contrast the two types of tools that come with OpenLDAP including those that operate directory on the LDAP database and those that use the LDAP protocol

1. `ldap*` commands communicate directly with the LDAP service, so they are not mediated by the DirectoryService layer. So DirectoryService will not perform the operation requested to generate some fundamental attribute or to create entries for PasswordServer or Kerberos. You will need to perform those operations manually.
   `ldap*` commands can add users to a running LDAP directory.
2. `slap*` commands operate directly on the LDAP data store.
   You need to shutdown the LDAP service when using slap tools.
   Changes you make with `slap*` commands do not get replicated to existing replicas.

Define the following term: Directory Access Controls (DACs)

1. DACs (also referred as LDAP ACLs) are used to enforce limited administrative privileges.
2. LDAP service evaluate each LDAP request against the DACLs t determine the level of authorization that it should extend to the request.

Describe how Mac OS X Server enforces Tired Administration

1. Tired Administration is when you assign to a standard user limited administrative capabilities over a group it's member of.
2. To enforce Tired Administration limits, Mac OS X uses DACs.

Define the following acronym: LDIF

LDIF stands for LDAP Data Interchange Format, which represents LDAP entries and changes records in text format. It is needed to operate with ldap tools.

Compare and contrast the possible uses of an LDIF file with those on an XML file exported from Workgroup Manager

1. LDIF files are used to import data with ldap commands, they can be used to import encrypted password.
2. XML files are used to import data with dsimport command, they cannot be user to import password.

Identify command-line OpenLDAP tools to add users to an Open Directory server

1. `$ dsimport <-a|-g|-s|-x> filePath DSNodePath <O|M|A|I|N> -u username -p password`
2. `$ ldapadd -H ldap://server.domain.name -f usersforldapadd.ldiff`
3. `$ slapadd -l usersforslapadd.ldiff`

Identify the log files used by Open Directory when promoting Mac OS X Server to an Open Directory Master

`/Library/Logs/slapconfig.log`

Describe the tools, process and files involved in the promotion of an Open Directory server

1. The promotion is handled by the slapconfig command:
   `$ sudo slapconfig -promotereplica diradmin`
2. The promotion process is logged to `/Library/Logs/slapconfig.log`
3. `$ /usr/bin/ldapmodify -c -x -H ldap://%2Fvar%2Frun%F2ldapi`
   Is executed to modify the entry "`olcDatabase={1}bdb,cn=config`"
   and "`cn=ldapreplicas,cn=config,dc=servername,dc=domain,cn=com`"
4. Slapd is then stopped & restarted (to apply the changes)
5. `$ /usr/sbin/kdcsetup -f /LDAPv3/127.0.0.1 -w -x -a diradmin -p **** -v 1`
   Is run to promote replica Password Server to master
6. `$ /sbin/kerberosautoconfig -u -v 1`
   Is run to update the directory records

Configuring Open Directory Replicas

Describe the topology of Open Directory replication

1. There is only one OD master.
2. There could be one ore more Tier 1 OD replicas which only duty should be to keep synchronization between OD master and Tier 2 replicas.
3. There could be one ore more Tier 2 replicas (to which the clients should connect) connected to Tier 1 replicas and NOT to the OD master.

If there are no Tier 2 replicas the client will be communicating directly with Tier 1 replicas.

Identify the maximum number of records supported in an OD master

200.000 records (for guaranteed efficiency).

Identify the amount of load an Open Directory client places on an OD master

1. 1000 connections on versions earlier than Mac OS X Server 10.6.
2. 8192 connections with Mac OS X Server 10.6.

Given an existing OD master and the Server Admin application in a standard installation of Mac OS X Server 10.6 create an Open Directory replica of the Open Directory master

1. You can do via command-line:
   ```
   $ sudo slapconfig -createreplica <odmaster IP address> diradmin
   ```
2. You can do via Server Admin, select Open Directory service and click 'Settings'.
3. Click 'General' tab and 'Change' button.
4. Select "Setup an Open Directory replica".
5. Fill the fields with:
   - IP address of the OD master (or Tier 1 replica).
   - Password for root account on the OD server/Tier 1 replica.
   - Domain administrator short name.
   - Password for the Directory Administrator.
6. Click 'Continue' and confirm the settings.

Given a failed Open Directory master and an Open Directory replica, recover from a failure of the Open Directory master

There are several options:

1. Promote the OD replica to be the new master.
   In this case after the promotion you must make all the relays and replicas standalone and then rebind them to the new master.
2. Create a new Open Directory master at the original IP address of the first OD master.
   You could configure a new Mac OS X Server computer with the old OD master's IP address and then you have to choose between:
   - Promote it to OD master and then restore from and OD archive.
   - Make it a replica of the new OD master (a temporary promoted replica), and then promote it to be the newest OD master.

After this you need to make all the replicas standalone and then make each one a relay or replica of the new OD master.

It's possible to promote a replica to master via Server Admin or via command-line with the command:

```
$ sudo slapconfig -promotetoreplica diradmin
```

Describe the tools, processes and files involved on creating an Open Directory replica

1. `/Library/Logs/slapconfig.log` contains the information from a replica creation.

2.
   ```
   $ ssh root@remoteip slapconfig -checkmaster diradmin 0 5 5
   ```
   is run to check that the conditions for a replication process are met.

3.
   ```
   $ ssh root@remoteip /usr/bin/slapconfig -stopldapserver
   ```
   stops the ldap service.

4.
   ```
   $ ssh root@remoteip /usr/bin/db_recover -h /var/db/openldap/openldap-data;
   $ ssh root@remoteip /usr/sbin/slapcat -l /var/db/openldap/openladap-data/backup.ldif
   ```
   Is run to restore the database to a consistent state and to create a LDIF file of the LDAP data store.

5.
   ```
   $ ssh root@remoteip /usr/sbin/slapconfig -startldapserver
   ```
   Starts the master LDAP server
6. Update master's configuration with:
   ```
   $ ssh root@remoteip /usr/sbin/slapconfig -adreplica youripaddress
   ```
   It appends a line to `/etc/openldap/ldap_macosxserver.conf`

7.
   ```
   $ /bin/rm -R /var/db/openldap/openldap-data
   ```

8.
   ```
   $ scp root@remoteip:/var/db/openldap/openldap-data/backup.ldif /var/db/openldap/openldap-data/
   ```

9.
   ```
   $ scp root@remoteip:/etc/openldap/schema /etc/openldap
   ```

10.
    ```
    $ scp root@remoteip:/etc/openldap/rootDSE.ldif /etc/openldap/rootDSE.ldif
    ```

11.
    ```
    $ /usr/sbin/slapdadd -c -l /var/db/openldap/openldap-data/back.ldif
    ```

12.
    ```
    $ /usr/sbin/slaptest -f /etc/openldap.conf -F /etc/openldap/slapd.d
    ```
    To generate a slapd.d configuration directory from slapd.conf and slapd_macosxserver.conf

13.
    ```
    $ scp root@remoteip:/etc/openldap/slapd.d/cn=config/cn=schema/cn={9}customSchema.ldif /etc/openldap/slapd.d/cn=config/cn=schema/cn={9}customSchema.ldif
    ```

14.
    ```
    $ /usr/bin/ldapmodify -x -c -H ldap://%2Fvar%2Frun%2Fldapi
    ```

15. `/var/db/authserver/authservermain.initial.gz` is generated with:
    ```
    $ ssh root@remoteip /usr/bin/mkpassdb -copy -gzip
    ```

16.
    ```
    $ scp root@remoteip:/var/db/authserver/authservermain.initial.gz /var/db/
    authserver/authservermain.gz
    ```

17.
    ```
    $ ssh root@remoteip /bin/rm /var/db/authserver/authservermain.initial.gz
    ```

18.
    ```
    $ /usr/bin/gunzip /var/db/authserver/authservermain.initial.gz
    ```

19.
    ```
    $ /usr/sbin/mkpassdb -soy -s 521 -e 1021 -n ReplicaX.Y
    ```
    Creates local Password Server.
20. `$ /usr/sbin/mkpassdb -setreplicationinsterval` *`n`*
21. `$ /usr/sbin/mkpassdb -setrealm` *`realm`*

22.
    ```
    $ /usr/sbin/mkpassdb -key
    ```
23. Enable local Kerberos server with:
    ```
    $ /usr/bin/kdcsetup -c /LDAPv3/127.0.0.1/ -a diradmin -p obscuredpasswd -v 1 REALM
    ```
    Force an update to `/Library/Preferences/edu.mit.Kerberos`

24.
    ```
    $ /usr/sbin/krb5_util -a REALM load /var/db/krb5kdc/initial.dump
    ```

25.
    ```
    $ /usr/sbin/kdcsetup -e
    ```
    To enable `kdcmond` and `kdcmind` in the configuration for launchd.

26.
    ```
    $ /usr/sbin/sso_util configure -x -r REAL -f /LDAP/127.0.0.1 -a diradmin -p
    hidden-password -v 1 all
    ```
    To create service principals for all the services that this server offers.

27.
    ```
    $ /sbin/kerberosautoconfig -u -v 1
    ```
    Force the update of `/Library/Preferences/edu.mit.Kerberos`.

28.
    ```
    $ /usr/sbin/vpnaddkeyagentuser -q /LDAP/127.0.0.1
    ```
    In case you ever want to use this server as VPN server.

Given the Console application review the logs involved in the Open Directory replication process

1. `/Library/Logs/PasswordService`
2. `/Library/Logs/slapconfig.log`

Connecting Mac OS X Server to Open Directory

Given the server admin application in a standard installation of Mac OS X 10.6 bind the server to an Open Directory master

1. Select Open Directory Server.
2. Click the 'Settings' button in the toolbar.
3. Check the 'General' tab.
4. Click the 'Change' button by the 'Role' description.
5. Select 'Connect to master directory' in the Open Directory Assistant.
6. Complete the process and click the 'Open Directory' button.
7. Bind to an Open Directory master or replica.
8. Verify that the OD master/replica is in the authentication path.

Given Server Admin application in a standard installation of Mac OS X 10.6 bind the server to an Open Directory Kerberos realm

1. In Server Admin select the OD server used for authentication and bin to it.
2. In the 'General' tab of the 'Settings' button in the toolbar, click the 'Join Kerberos' button.
3. In the 'Realm' pop-up menu in the 'Join Kerberos Realm' dialog choose the share Open Directory realm and provide credentials for a network administrator who has the ability to join a computer to the Kerberos realm.
4. The 'Join Kerberos' button in Server Admin should disappear as sign that you are bound to the Kerberos Realm.

List services which can use network user accounts for access control

1. AddressBook
2. AFP
3. FTP
4. iCal
5. iChat
6. Mail
7. SMB (CIFS)
8. VPN
9. Xgrid
10. Login Window
11. Podcast Producer
12. Quicktime Sharing
13. Blog
14. RADIUS

Describe the use of UUID (also known as GUID) to identify users in an Open Directory system

1. When you add an OD master user or group to either a SACL of file system ACL the operating system uses the UUID (GUID) in addition to or instead of the record's name.
2. If the connection of Mac OS X Server to its OD system is interrupted you will see the 128-bit unique identifier listed instead of the record name.

Given the Workgroup Manager application and Mac OS X Server bound to an OD master view users on the parent OD master

1. In WGM choose 'Preferences' from the menu.
2. Ensure that 'All Records' tab and 'Inspector' checkbox are selected.
3. Click the "globe" icon and select your shared domain or select 'Other' and navigate through LDAPv3 to the server for your shared domain.
4. In the toolbar click the 'Accounts' button.
5. In the column click the 'Users Account' icon to list all the available network users at the domain.

Describe how keytabs and principals are created when joining a kerberos realm with Server Admin

The Server Admin's 'Join Kerberos' button is equivalent to the command-line `sso_util` command.

When you join your Mac OS X Server computer to a Kerberos realm, `sso_util` sets up a service principal with randomly generated keys for each of your services that can be kerberized.

The KDC stores principals in the principal-data store in `/var/db/krb5kdc/principal.REALM and` the newly joined server stores the principals in `/etc/Krb5.keytab`.

Define the following terms: keytab, principal

1. Keytab is a short for Keytable and it is a collection of key or a collection of principals and their keys.
2. A Kerberos principal is a Kerberos entity. Each principal has a key associated with it which is a secret string of characters. There exist service principals, user principals and host principals (for the `host` service).

Given the Console application on a Mac OS X server which has joined an Open Directory master, view the log files that document the binding process

Consult:

1. /Library/Logs/slapconfig.log
2. /var/log/krb5kdc/kadmin.log
3. /var/log/krb5kdc/kdc.log

Describe the tools, processes and files used in joining a Kerberos realm

The `sso_util` command (Single-Sign-On utility) is used to join a Kerberos realm.

Logs files updated (`$ sudo sso_util info -L`):

- Admin_server: `FILE:/var/log/krb5kdc/kadminlog`
- KDC: `FILE:/var/log/krb5kdc/kdc.log`

1. The command contact the directory server and looks for the realm (attribute: apple-config-realm;cn=KerberosKDC,cn=config,searchbase).
2. The command then creates a list of service to kerberize: ("All" includes: cifs,ldap,xgrid,van,pip,xmpp,host,smtp,nfs,http,pop,imap,ftp).
3. For each service the command creates service principals and keys through `kadmin` command
   ```
   $ kadmin -r $REALM -p$DIRADMIN -w $DAPASSWD -q "add_principal
   -randy service/$ODMASTER@$REALM" -s $ODMASTER
   ```
4. "`sso_util configure`" uses "`kadmin ktremove`" to delete all existing and potentially outdated entries for each service in the realm from the local `/etc/Krb5.keytab`
   ```
   $ kadmin -r $REALM -p$DIRADMIN -w $DAPASSWD -q "ktremove -k /
   etc/Krb5.keytab service/$ODMASTER@$REALM all" -s $ODMASTER
   ```
5. The command then adds the principals you just generated to the keytab file on the local server using "`kadmin ktadd`".
6. "`sso_util configure`" configures services to use the new kerberos realm with the `krbservicesetup` command:
   ```
   $krbservicesetup -r $REALM -a $DIRADMIN -p $DAPASSWD -t /etc/
   Krb5.keytab -f /tmp.RANDOM/setup
   ```

   Krbservicesetup updates the configuration files for these services:

   - VPN: `/Library/Preferences/SystemConfiguration/com.apple.RemoteAccessServices.plist`
   - CIFS:
     - `/System/Library/LaunchDaemons/nmdb.plist`
     - `/System/Library/LaunchDaemons/smbd.plist`
     - `/System/Library/LaunchDaemons/org.samba.winbindd.plist`
     - `/etc/smb.conf`
     - `/Library/Preferences/SystemConfiguration/com.apple.samba.server.plist`
     - `/var/db/smb.conf`
   - FTP:
     - `/Library/FTPServer/Configuration/ftpaccess`
     - `/System/Library/LaunchDaemons/xftpd.plist`
   - Xgrid: `/etc/xgrid/controller/service-principal`
   - IMAP, POP and SMTP: `/etc/MailServicesOthers.plist`
   - AFP: `/Library/Preferences/com.apple.AppleFileServer.plist`

   1. "`sso_util configure`" uses `kerberosautoconfig` to update or create the Kerberos configuration file in `/Library/Preferences/edu.mit.Kerberos` and the DSLocal node on `/var/db/dslocal/Default/config/Kerberos:REALM.plist`.

Given Directory Utility on a Mac OS X server which has joined an Open Directory master revert the server to a pre-join state and join again

Change your server back to standalone status:

1. Open Server Admin and select Open Directory from the list of services, then click the 'Settings' button and as last the 'General' tab.
2. Click the 'Change' button.
3. In the Open Directory choose 'Type' window, select 'Disconnect and set up standalone directory' and click 'Continue'.
4. In the 'Config Settings' window, click 'Continue'.
5. Close the 'Server Configuration Complete' window.

Connect your server to a directory system again:

1. Open the Directory Utility application.
2. Click the Services button (if necessary).
3. Click the lock icon and authenticate (if necessary).
4. In the toolbar, click 'Search Policy', and then click the 'Authentication' tab.
5. Change the search path from 'Local Directory' to 'Custom Path' and click 'Apply'.
6. Click 'Apply' and then quit Directory Utility.
7. Principals and keytabs are already created and stored in /etc/krb5.keytab and your services are still configured to accept Kerberos authentication.

Explain how a TGT is generated and used for authentication

1. The client sends the KDC a Kerberos authentication service request (KRB_AS_REQ).
   It includes:
   - User principal.
   - Service name (krbtgt).
   - The requested time constraints on the tickets.
   - Possible encryption types.
2. The KDC rejects the request with ERR_PREAUTH_REQUIRED.
3. The client generates preauthentication data in the form of a timestamp, encrypted with the user's password.
   The client sends another KRB_AS_REQ to the KDC with preauthentication data included
4. The KDC (which stores the user's principal's password in /var/db/krb5kdc/principal) decrypt the preauthentication data. The KDC consider the user as authenticated.

   The KDC generates a session key to encrypt communications between the client and the KDC (a shard secret).

   The KDC generates a TGT, which it will send to the user.

   The Client is not able to decrypt the TGT because it's encrypted with a private key that only the KDC knows.

The TGT contains:

- KDC's copy of the session key.
- The Client principal's name.
- The ticket time constraints.
- A timestamp.
- The client's IP address (optional).

The KDC sends a Kerberos authentication service reply (KRB_AS_REP) to the client that contains:

- The session key encrypted with the user's password.
- The TGT.

5. The client saves the TGT and the session key in the client's credential cache.
   IT IS STORED EXCLUSIVELY IN THE RAM to make difficult to attackers to obtain TGT or session key.

   The client decrypts the session key using its password to be used to encrypt communications.
   The client sends the KDC a Kerberos ticket-granting service request (KRB_TGS_REQ) that includes:

   - The TGT.
   - The realm.
   - The principal name for the requested service.
   - The requested ticket life time.
   - The supported encryption types.
   - Preauthentication information: a timestamp encrypted with session key.

6. The KDC decryption the preauthentication information and the TGT included in the KRB_TGT_REQ

THE USER IS AUTHENTICATED!

Explain how a Kerberos service ticket is generated and used for authentication

1. The KDC generates a random session key that the client and the service will use to communicate with each other.

   The KDC prepares a service ticket that contains:

   - The service session key.
   - The user principal.

   The KDC encrypts the service ticket with the service's key (only the service and KDC can decrypt the packet).

   The KDC prepares the Kerberos Ticket-granting service replay (KRB_TGS_REP) and sends it to the client. It contains:

   - The new service session key (encrypted with the user's password).
   - The service ticket (encrypted with the service's private key).

2. The client generates an authenticator: a timestamp encrypted with the service session key.

   The client sends a Kerberos application request (KRB_AP_REQ) to the server hosting the service that the client wants to use. It includes:

   - Service ticket.
   - The authenticator.

   OR

   The client makes a request for service with the service ticket embedded in the request.

3. The server that hosts the desired service attempts to decrypt the service ticket and the authenticator.

   It uses the service key to decrypt the service ticket.

   Inside the service ticket is the service session key which the service uses to decrypt the authenticator.

   THE USER IS AUTHENTICATED!

Define the following terms: service principal, user principal

1. A service principal's name defines:
   - The service type.
   - The server hosting the service.
   - The Kerberos realm.
2. A user principal's name is usually (user@SERVER.DOMAIN.COM):
   - Just the user's shortname.
   - The '@' sign.
   - The REALM.

Given a Mac OS X 10.6 computer bound to an OD master, verify the service principals on the server are valid.

1. Open Terminal on the Mac OS X Server computer you joined to the Kerberos realm.
2. Use klist -ke to get a list of all the principals in the default keytab:
   ```
   $ sudo klist -ke [|grep <service name>|grep -v KDC]
   ```
3. Use kadmin to connect to the remove KDC and get information about principal
   ```
   $ sudo kadmin -r $REALM -p $DIRADMIN -w $DAPASSWD -q
   “get_principal servicename/server.domain.com@$REALM”
   ```

Confront the values for the KVNP and encryption types from step 2 and step 3 if they match the principals are valid.

Describe how setting up and using a network time server can help avoid some Kerberos authentication issues

1. If the system clocks are more than 5 minutes apart, after considering time zones, you cannot authenticate using kerberos.
2. You should use NTP to synchronize the clocks of all your computers.
3. If you cannot access a public NTP server you can use one of your Mac OS X Server computers as NTP server.
4. On the server you want to make your local NTP server change to `time.apple.com` to `127.127.1.0`. This forces the NTP service to use the local clock.

Integrating Mac OS X Server with Other Systems

Define the following term: augmented record

An augmented user record (or just augment) is a record that allows you to add a specific set of extra attributes to an existing record from an external directory.

You can use only another Open Directory or Active Directory node as a source of original users to “import” for augmented user records.

Describe how augmented records can extend the schema of a third party directory service

To let Directory Service return the attributes of an augment as result of a query you need to manually edit the augment certification record (`cn=augmentconfiguration,cn=config,searchbase`).

This record does not exist in the Open Directory LDAP database automatically.

When you use Workgroup Manager to create an automount record for a share point OD creates a record named `augmentconfiguration` in `cn=config,searchbase`.

This record contains an attribute `dsAttrTypeStandard: XMLPlist` that defines the attributes that DirectoryService returns for the augment.

You will modify the XMLPlist attribute to allow other attributes.

The XMLPlist contains:

- A list of attributes that Open Directory uses to build the augment.
- The node that contains the augment records: `/LDAPv3/server.domain.com`
- The node that contains the original records: `/Active Directory/All Domains`

Given a Mac OS X computer bound to multiple directories, identify which directory will be used for identification

Open the Open Directory utility and verify the Authentication Search Path.

Directory Service looks for a matching user record in each mode in the authentication search path and then stops searching with the first match.

Given a Mac OS X computer bound to multiple directories, identify which directory will be used for authentication

The user record’s AuthenticationAuthority determines how the system will attempt to authenticate the user. If there is no AuthenticationAuthority attribute, DirectoryService will attempt to guess the user’s kerberos principal based on the local Kerberos configuration and will authenticate with Kerberos.

As a last resort, DirectoryService will attempt to authenticate the user via an LDAP bind against Th. LDAP server that hosts the user record.

List the object classes Open directory adds to the standard LDAP schema

1. Container Structural Object Class
2. Time to Live Object Class
3. User Object Class
4. Group Auxiliary Object Class
5. Machine Auxiliary Object Class
6. Mount Object Class
7. Printer Object Class
8. Computer Object Class
9. Computer List Object Class
10. Configuration Object Class
11. Preset Computer List Object Class
12. Preset Computer Object Class
13. Preset Computer Group Object Class
14. Preset Group Object Class
15. Preset User Object Class
16. AuthenticationAuthority Object Class
17. Server Assistant Configuration Object Class
18. Location Object Class
19. Service Object Class
20. Neighborhood Object Class
21. ACL Object Class
22. Resource Object Class
23. Augment Object Class
24. Automount Map Object Class
25. Automount Object Class

Describe the role of keytabs in kerberos authentication

The keytab contains the shared secret between the KDC and the service.

Given a computer with Mac OS X Server installed and an existing directory server, configure the Mac OS X server to augment the user records of the existing directory server

1. On the server setup an automount record for a network home folder share point (to generate the `augmentconfiguration` record).
2. Open WGM, connect to your OD master edit /LDAP/127.0.0.1 mode use 'Preferences' to select "Show 'All Records' tab and Inspector'.
3. Click the 'All Records' button.
4. Click the pop-up menu under the search field and choose 'Config'.
5. Choose the `augmentconfiguration` record.
6. Highlight the XMLPlist attribute and click 'Edit'.
7. Edit the 'Augment Attribute List' as preferred.
8. In WGM create a new augment user record.
9. Click the Users button to edit an augmented user record.
10. Click the Inspector tab.
11. Select the newly created attribute and modify its value.
12. Click the 'New Attribute' and choose among the new type of attribute just created in the XMLPlist, the modify it's content.
13. Click 'Save' to save the edits to the augment user record.

Given Kerberos tools in a standard installation of Mac OS X Server 10.6 configured as an OD master, create service keys for third-party kerberos service

1. Create a principal in the third-party Kerberos realm for each Mac OS X Server service.
2. Create a keytab for each Mac OS X service (to contain the principal).
3. Extract the keytabs into a keytab file to be copied to the Mac OS X server.
4. Copy the keytab file to the Mac OS X Server computer.
5. Import the keytab entries into a Mac OS X Server's keytab file /etc/krb5.keytab
   - `$ ktutil`
   - `ktutil: read_kt /etc/krb5.keytab`
   - `ktutil: list`
   - `ktutil: read_kt /[...]/keytabforosx.keytab`
   - `ktutil: list`
     `(will show the merged content of both files)`
   - `ktutil: write_kt /etc/krb5.keytab`
   - `ktutil: quit`
6. Remove the first 4 lines from /Library/Preferences/edu.mit.Kerberos file so to disable the automatic configuration.
7. Prepare your Mac OS X Server services to use the new Kerberos realm

```
$ sudo /usr/libexec/PlistBuddy -c “Set kerberosPrincipal <servicename>/
server.domain.com@REALM” /Library/Preferences/com.domain.ServiceName.plist
```