Unified Windows® and UNIX® Authentication Using Microsoft® Active Directory Kerberos

Ellie Berriman, Network Appliance, Inc.
May 2006, TR-3457

Abstract

This guide will provide guidance on configuring a unified Windows and UNIX authentication solution for NetApp storage systems based on Microsoft Active Directory Kerberos. The goal is to provide all information needed to successfully integrate a NetApp storage system into the environment. The actual customer environment can and probably will vary from the scenario provided in this guide. However, this guide's goal is to provide the necessary understanding that will allow the customer to successfully configure the storage system into the actual infrastructure.

Table of Contents

1. Introduction

1.1 Overview of Kerberos in a Heterogeneous Environment

Kerberos is a protocol, defined in RFC 1510, designed to provide strong authentication within a client/server environment. The basis of the protocol is a shared secret key cryptology system. MIT created the Kerberos authentication model in the early 1980s as a way of providing secure authentication in a networked environment.

Kerberos uses shared key encryption to ensure the confidentiality (no inappropriate access to data) of the data and uses hashing techniques to ensure the integrity of the data (no ones has messed with the data).

During the last several years Kerberos has been gaining acceptance as a secure, network-based authentication service. Many companies are replacing local system authentication with Kerberos authentication.

Microsoft implemented Kerberos as the primary authentication service in its Windows 2000/2003 Active Directory. The Microsoft Kerberos implementation is standards-based, resulting in the ability to use Microsoft Active Directory Kerberos for UNIX and Linux® Kerberos authentication. This provides a method to unify authentication on networks based on UNIX and Windows.

With the Network Appliance™ multiprotocol storage platform, where clients based on UNIX or Windows can access data using CIFS or NFS, providing the ability to use standard network services for authentication and for identity storage is crucial. Data ONTAP® has historically supported integration into an Active Directory environment for Windows user authentication and authorization. More recently, the ability to use Active Directory Kerberos for UNIX authentication and AD LDAP as a directory store for UNIX user and group information has been provided as well.

.

1.2 Scope and Purpose of this Guide

This report describes an environment that utilizes Microsoft Active Directory Kerberos. This report will guide the reader through the configuration of the entire infrastructure needed to support unified Kerberos authentication in a NetApp storage system environment. Although configuration of the storage system to use Active Directory Kerberos for NFS is only one small part of the process needed to implement a Kerberos solution in the enterprise, we will step through the entire process so that the role of Kerberos on the storage system and its integration into the entire environment becomes clear.

This guide is not intended as a definitive method to set up a unified Kerberos solution in the customer environment. There are a of number Kerberos implementations available. Additionally, there are many different UNIX and Linux client operating systems running a number of versions, with many different configurations. Therefore, it is expected and likely that any customer environment will differ in some ways or differ significantly from the one described in this report.

However, this report will give guidance on the principles of how all components work together. This report will also give substantial information on the use of Kerberos on the NetApp storage system. The goal of this report is to provide all necessary information needed to configure the storage system to use Kerberos for UNIX authentication. By stepping through the entire process, the goal is to provide a knowledge base that can be used during storage system integration into the actual customer environment.

After an introduction to Microsoft Active Directory as a unified solution for authentication and authorization, this guide will follow these steps to configure the environment:

- Infrastructure Preparation
- Configure and enable Kerberos on the storage system.
- Integrate a Solaris™ 9 client to use Microsoft Active Directory Kerberos authentication.
- Configure the storage system and the UNIX client to allow NFS v3 mounts with Kerberos security.

Appendix A provides configuration information for configuring a Fedora 4 client to use Microsoft Active Directory Kerberos authentication.

2. Introduction to Unified Authentication and Authorization Using Windows Active Directory

There are two main hurdles a user must clear before being able to exercise rights on a system or access data. The user must first be authenticated and then must be authorized to perform the desired action, whether it is rebooting the system, using a printer, or accessing a file.

2.1 Authentication

Authentication is the process of verifying the identity of an entity: that is, verifying that the entity is who he or she claims to be. This is important in ensuring the confidentiality, integrity, and availability of information.

Authentication can be based on something the entity knows, such as a password; an external object, such as smartcard; or biometrics such as a retinal scan. The configurations used in this tech report will use the password method.

Authentication depends on an authentication service. Windows AD domains natively use Active Directory integrated Kerberos authentication. Many UNIX environments are now using Kerberos authentication, which is a secure, network-based system. The configuration presented in this report is using Kerberos as the primary authentication method on the UNIX hosts.

2.2 Authorization

Authorization is the process of determining what an authenticated entity is able to do. Authorization includes rights on the system as well as rights on the data on the system. Authorization as it relates to data determines if an entity has a right to read, modify, delete, or change the permissions on files and folders.

On NTFS file systems, NTFS permissions determine what various entities can do with the file or folder. On UNIX file systems, the UNIX style file permissions “rwxrwxrwx” determine what the owner, the group, and others can do on that file or directory.

The processes of authentication and authorization work together to maintain appropriate controls on the access of data and the exercise of rights in the enterprise.

To see further information on using Microsoft AD LDAP as a directory store please refer to TR3458, “Unified Windows and UNIX Authorization Using Microsoft Active Directory LDAP as a Directory Store.”

2.3 Authentication Services with Kerberos

MIT created the Kerberos authentication model in the early 1980s as a way of providing secure authentication in a client/server environment. Starting with Windows 2000, Kerberos is the primary, preferred method of authentication within a Windows domain environment. Windows 2003 implements Kerberos V5 as a security support provider, which is a dynamic-link library loaded by the local security authority at boot.

Since a Microsoft implementation is standards-based, the same Kerberos SSP can provide authentication services for UNIX and Linux hosts as well.

The basis of the protocol is a shared secret key cryptology system. Kerberos uses shared key encryption to ensure the confidentiality (no inappropriate access to data) of the data and uses hashing techniques to ensure the integrity of the data (no ones has messed with the data).

2.4 Kerberos Components and Terms

The following components are present in a Kerberos environment

- **Realm:** A Kerberos realm is the set of Kerberos principals that are registered within a Kerberos server.
- **Principal:** The term used to refer to every entity within the Kerberos database.

  Users, computers, and services running on a client are all principals. Every principal is unique within the Kerberos database and is defined by its distinguished name. A principal name has three parts.

  primary/instance@REALM

  - **The primary:** Can be a user or a service. The primary can be a service such as the "nfs" service. It can also be the special service "host," which signifies that this service principal is set up to provide various network services such as ftp, rsh, nfs, etc.
  - **The instance:** This is optional in case of a user. A user may have more than one principal. For example, Fred may have a principal that is for everyday use and a principal that allows privileged use such as a sysadmin account. The instance is required for service principals and designates the FQDN of the host providing the service.
  - **The realm:** A Kerberos realm is the set of Kerberos principals that are registered within a Kerberos server. By convention, usually the realm name is the same as the DNS name, but converted to capital letters. Capital letters are not obligatory, but the convention allows easy distinction between the DNS name and the realm name.
  - **Examples:**
    fred@ELLIE.COM
    fred/admin@ELLIE.COM
    host/saturn.ellie.com@ELLIE.COM
    root/saturn.ellie.com@ELLIE.COM
    nfs/eddie.ellie.com@ELLIE.COM

- **Ticket:** The fundamental unit of Kerberos authentication. Tickets are passed between the client, the KDC, and the server offering services.
- **Key distribution center:** The KDC has the following components:
  - A database of principals, containing users, computers, and services
  - An authenticating server; responsible for granting ticket granting tickets (TGTs)
  - A ticket granting service (TGS): responsible for granting service tickets that grant clients access to services

2.5 The Kerberos Model

The KDC is implemented as a single process with two services:

- **Authentication service:** Before getting tickets to network services, a principal must first get a TGT from the authentication service.

- **TGS:** Issues tickets good for admission to other services in the domain.

  A TGT is valid for requesting services only within that Kerberos domain. If a Kerberos entity is requesting admission to the TGS of a trusted domain, an intermediate step is required. If the user does not have a TGT that is good within that domain, he must first get a referral TGS from the KDC TGS in his own domain that can be used as a referral to the trusted domain's TGS.

2.6 Active Directory Kerberos

Kerberos V

Kerberos V, the latest version of Kerberos, used in the Microsoft implementation, supports several new features important in Active Directory Kerberos.

- **Credential forwarding:** If the client connects to an application server, the user's TGT credentials can be forwarded to the application server for use on that server. The user can transparently authenticate to other Kerberos services from the application server.

- **Multiple encryption types:** Microsoft Active Directory Kerberos supports two of the encryption types, "des-cbc-crc" and "des-cbc-md5."

- **Renewable tickets:** The ticket may be renewed by resubmission to the KDC. Active Directory's implementation has a default renewable lifetime of 10 hours.

- **Preauthentication:** This involves the principal proving the user's identity before the KDC issues the ticket. This is done by sending a time stamp encrypted with the user's key. Active Directory KDCs require preauthentication. This is an added security feature that makes password guessing attacks difficult.

- **Support for GSS-API:** Active Directory Kerberos supports the use of Generic Security Service Application Programming Interface (GSS-API), described in:
  - RFC 1510, "The Kerberos Network Authentication Service (Version 5)"
  - RFC 1964, "The Kerberos Version 5 GSS-API Mechanism"

Active Directory Realms

UNIX based Kerberos implementations maintain a master copy of the principal database on the master KDC and use the Kerberos replication method to maintain copies of the master database on other designated KDCs. Microsoft Active Directory integrated with Kerberos uses a multimaster model. Every domain controller has its own KDC service, with replication being maintained by Active Directory replication rather than Kerberos replication.

- **krbtgt:** Krbtgt is the security principal used by the AD KDC. It cannot be deleted or renamed. The password for this account is used to derive a secret key for encrypting and decrypting TGTs that it issues. Clients address messages to a domain KDC by including both the SPN krbtgt and the name of the domain.

- **Locating the Kerberos KDC:** When a client computer needs to request Kerberos credentials, it needs the IP address of the KDC server. This is information is stored in DNS as service records.

- **TCP and Kerberos:** Windows 2003 authorization data can easily exceed the size of a single UDP packet, making Kerberos communication with UDP problematic. The newest proposal is to allow TCP transport to carry Kerberos traffic. This has been implemented in the latest MIT Kerberos versions (MIT 1.3.1 or later). See Section 2.6.1 below for more information on authorization data within the Kerberos packet.

Kerberos Policy

Within the Windows domain Kerberos policy is implemented through the default domain group policy. Kerberos policy includes:

- **Enforce user logon restrictions:** When this policy is enabled the user requesting the session ticket must have the right to either “Log on Locally” or “Access this Computer from the Network.”
- **Maximum lifetime for service ticket:** The maximum amount of time that a ticket granted for a service (a session ticket) can be used. Default is 600 minutes.
- **Maximum lifetime for user ticket:** The maximum amount of time that a user’s TGT can be used. Default is 10 hours.
- **Maximum lifetime for user ticket renewal:** The longest period of time that a TGT can be used if repeatedly renewed. Default is 7 days.
- **Maximum tolerance for computer clock synchronization:** The maximum difference in minutes that Kerberos will allow the time of the client’s clock and the Kerberos server’s clock to differ. Default is 5 minutes.

2.6.1 Kerberos and Authorization in Active Directory Domains

Kerberos was designed for authentication purposes, not authorization purposes. However, the current Kerberos implementations have extensions that assist with authorization by providing a field for authorization data in Kerberos tickets. Therefore, the Microsoft Kerberos authentication reply contains user and group membership information in the Kerberos ticket. When the Kerberos protocol is used for authentication, a list of SIDs identifying a security principal and the principal’s group membership is transported to the local computer in the authorization data field of a session ticket.

Authorization data is gathered in two steps:

- The first step takes place when the KDC in a Windows Server 2003 domain prepares a TGT.
- The second step is accomplished when the KDC prepares a session ticket for a server in the domain.

When a user requests a TGT, the KDC queries AD for the user’s SID and the SIDs of all the groups that the user belongs to. If this is a multidomain environment, the KDC also queries the GC for any universal groups that include the user or one of the user’s groups. All the SIDs are placed in the authorization field of the TGT.

When the user requests a session ticket for a server, the KDC in the server’s domain copies the contents of the TGT’s authorization data field to the session ticket’s authorization data field.

If the server’s domain is different from the user’s account domain, the KDC queries AD to find out whether any security groups in the local domain include the user or one of the user’s groups. If so, the groups SIDs are added to the list in the session ticket.

2.7 The Windows Logon Session and Request for Services

The following process is followed when a Windows user logs onto the domain and requests services:

1. **The user asks for admission to the TGT for the domain.**
   This is an authentication service exchange between the Kerberos SSP and the KDC on the user's domain (KRB_AS_REQ and KRB_AS_REP).

   The result is a TGT that the user can use to request session keys to services.

2. **The user uses the TGT to ask for admission to the TGS for the domain in order to request a TGS ticket for the computer.**
   This is a TGS exchange between the Kerberos SSP on the computer and the KDC for the computer's account domain (KRB_TGS_REQ and KRB_TGS_REP).

   The result is a session ticket that the user can present when requesting access to system services on the computer.

3. **The user asks for admission to local system services on the computer.**
   This happens when the Kerberos SSP on the computer presents a session key to the LSA on the computer.

4. **Now that the user is logged onto and has access to resources on the local computer, the user uses the TGT to ask for admission to the TGS for the domain in order to request a TGS ticket for other services (such as CIFS).**
   This is a TGS exchange between the Kerberos SSP on the computer and the KDC for the remote system's account domain (KRB_TGS_REQ and KRB_TGS_REP).

   The result is a session ticket that the user can present when requesting access to system services on the remote computer or storage system.

5. **The user asks for admission to the desired services.**
   This happens when the Kerberos SSP on the computer presents a session key to the LSA on the computer offering the requested services.

Note: If the computer is not in the same domain as the user's account another step occurs. Before requesting a session ticket for the computer, the Kerberos SSP must first ask the KDC in the user's account domain for a TGT good for admission to the KDC in the computer or storage system's domain. It then sends that TGT to the storage system's domain KDC in order to get a session ticket for the storage system.

2.8 UNIX Authentication and Request for Access to Services

When a UNIX or Linux Kerberos principal needs access to services the principal must first receive a TGT, which is then used to request a TGS ticket for the service. The two-tier process follows the steps outlined below.

Request the TGT:

1. A client (a user or a service such as the host service) begins a Kerberos session by requesting a ticket-granting ticket (TGT) from the key distribution center (KDC). This request is often done automatically at login.
2. The KDC creates a ticket-granting ticket and sends it back, in encrypted form, to the client. The client decrypts the ticket-granting ticket by using the client's password.
3. Now in possession of a valid ticket-granting ticket, the client can request tickets for all sorts of network operations, such as rlogin or telnet, for as long as the ticket-granting ticket lasts. This ticket usually lasts for a few hours. Each time the client performs a unique network operation, it requests a ticket for that operation from the KDC.

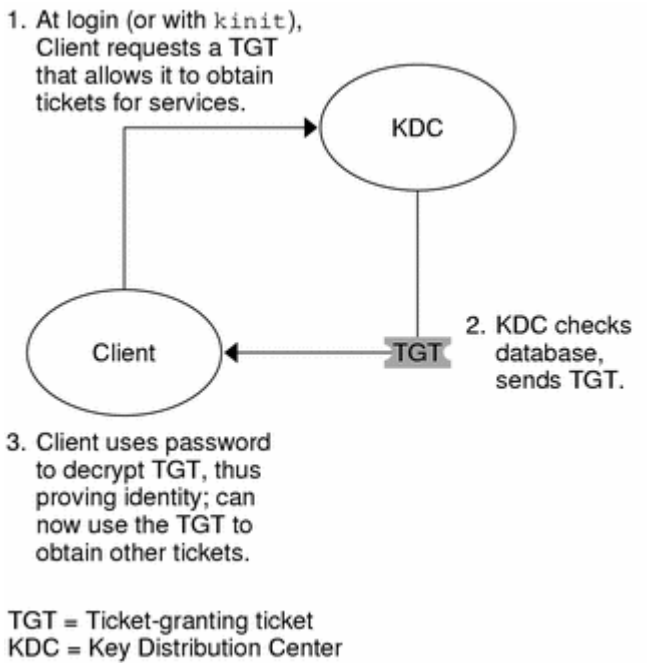


Figure 1) Initial authentication for Kerberos session.

Request the TGS:

1. The client requests a ticket for a particular service by sending the KDC its ticket-granting ticket as proof of identity.
2. The KDC sends the ticket for the specific service to the client.

   For example, suppose user “Pete” wants to access an NFS file system that has been shared with krb5 authentication required. Since he already has a ticket-granting ticket, as he requests access to the files, the NFS client system automatically obtains a ticket from the KDC for the NFS service.
3. The client sends the ticket to the server.

   When requesting access to data, the NFS client automatically and transparently sends the ticket for the NFS service to the NFS server.
4. The server allows the client access.

.

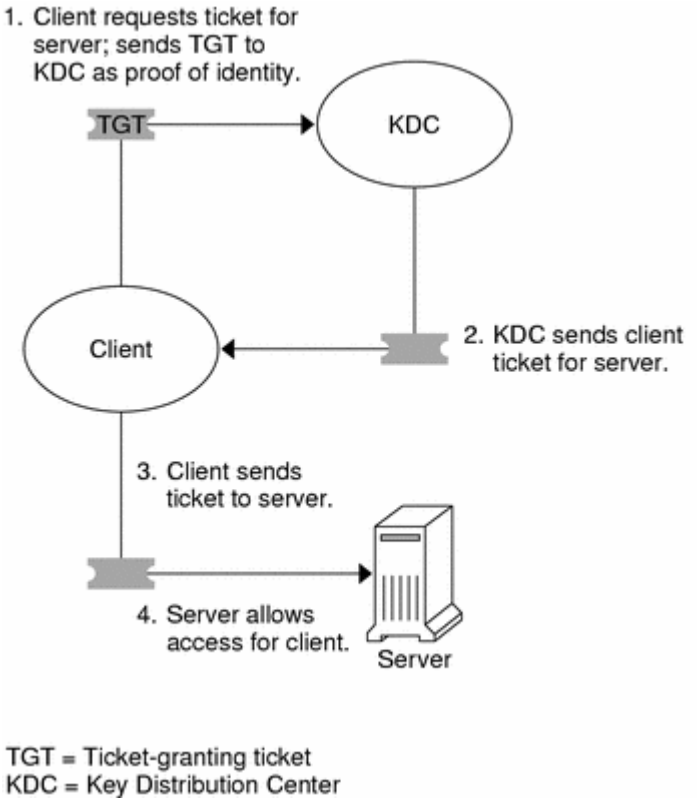


Figure 2) Obtaining access to a service.

3. UNIX Client Directory Store and Authentication Mechanisms

Just as Microsoft Active Directory uses one set of services for directory service and another set of services for authentication, UNIX clients have separate mechanisms for identity storage and for authentication.

3.1 Directory Stores

The following directory services are typically used on UNIX clients for storing information used for authentication and authorization. The configuration of which services are used and in what order is configured in the /etc/nsswitch.conf file.

- Local files
- NIS
- NIS +
- LDAP

3.2 Name Service Switch

Name service switch is an architecture that was defined to provide a convenient method for choosing methods of providing UNIX or Linux configuration information. This defines an interface between the standard C programming function calls and a service module that implements the storage of UNIX information, including user and group data used in authentication and authorization.

The following NSS databases are of particular interest when using AD Kerberos/LDAP for a unified authentication and authorization system:

- hosts
- passwd
- shadow
- group

The configuration file "**/etc/nsswitch.conf**" defines what name service method is used to retrieve information from each database. A configuration entry begins with the name of the database suffixed with a colon and ends with a list of name services to use, in order of their preferred use.

```
hosts:      files   dns   ldap   nis
passwd:     files   ldap  nis
shadow:     files   ldap  nis
group:      files   ldap  nis
```

The LDAP_NSS name service module must be installed on the UNIX or Linux system before LDAP can be used for name services in the nsswitch.conf file. Most modern UNIX or Linux distributions will have this module installed by default. However, some older versions of NSS_LDAP do not contain needed code to allow Microsoft Active Directory to be used as a UNIX LDAP directory store. For UNIX installations that contain versions of NSS_LDAP that do not support Active Directory LDAP, newer versions must be installed on the UNIX client.

This report uses Active Directory LDAP as the primary directory store for UNIX user and group information, but does not delve into the method and steps needed to implement this solution. For more information on implementing AD LDAP as a unified solution, please see TR 3458, "Unified Windows and UNIX Authorization Using Microsoft Active Directory LDAP as a Directory Store."

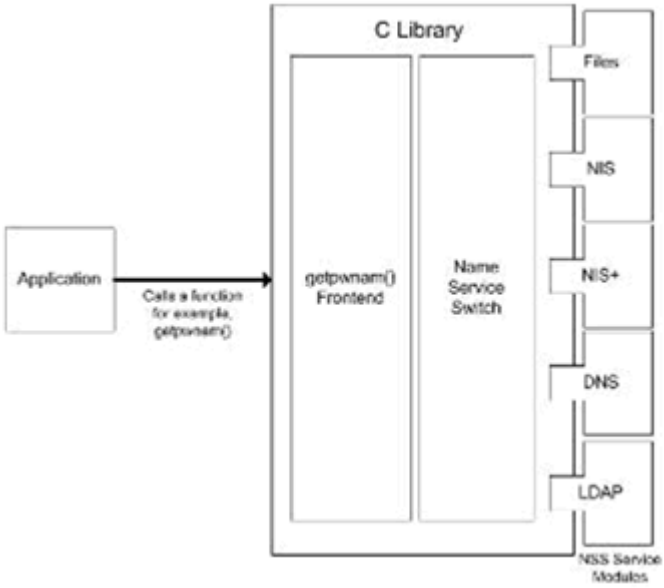


Figure 3) Name service switch service modules.

3.3 Authentication Services

The following authentication services are typically used on UNIX clients. The PAM configuration file is used to define which of the services are used, what order they are used, and what happens in event of the success or failure of each authentication service.

- Local system
- Kerberos based on UNIX Kerberos servers
- Active Directory–based Kerberos
- LDAP (not recommended for user authentication in an enterprise environment; generally used to authenticate binds to the LDAP directory)

3.4 Pluggable Authentication Modules

Pluggable authentication modules, or PAM, is a framework that provides a method to “plug in” a variety of authentication methods for authenticating users on UNIX or Linux clients. Every PAM represents a specific authentication mechanism. Each module in turn has one or more module types which defines what the module does.

There are four module types configured through the PAM configuration file:

- **Authentication**: Authenticates the user using a chosen authentication mechanism: system, Kerberos, LDAP, etc.
- **Account**: Provides services for checking the validity of the account: password aging, account disabled, access hour restriction, etc.
- **Password**: Provides methods for managing passwords.
- **Session**: Manages the opening and the closing of an authentication session. Also provides facility to define what needs to take place before a service being granted and after it is terminated.

On systems that use PAM, all logon processes and utilities that require user authentication must be configured to use PAM.

3.5 The PAM Configuration File

Authentication methodology through PAM is configured in the PAM configuration file. It is used to determine the authentication services to be used, what order the configured authentication services are used, and what to do at the success or failure of each configured authentication. For Solaris, the configuration file is /etc/pam.conf. Linux clients have a separate directory for PAM configuration, /etc/pam.d. Within the /etc/pam.d directory, there are individual files for each service. In addition to individual files for each service, the pam.d directory has a single file, /etc/pam.d/system-auth, used by the authconfig utility. Using the authconfig utility is a convenient way to configure all authentication services without manually editing individual files.

The PAM configuration file consists of a set of entries. Each line in the pam.conf file has the following format:

service_name module_type control_flag module_path module_options

- ***service_name***: the name of the service requiring PAM: login, rlogin, ftp, telnet, etc. The service name “other” allows a default to be set for any service not specifically configured within the file.
- ***module_type***: Account, authentication, session, password.
- ***control_flag***: Determines the continuation or failure behavior from the module. The following control flags are valid:
    - ***requisite***: If successful, record a required success and keep checking the other modules. If it fails, record a failure and stop any further checking. If a requisite fails, the overall request fails, regardless of the response of other modules. A success from a requisite module does not mean that the request will necessarily succeed. All the required modules must succeed before the overall request can succeed..
    - ***required***: If successful, record a required success and keep checking other modules. If this fails, keep checking other modules. A failure with this request will prevent success even if other required modules are successful.
    - ***optional***: If successful, record an optional success and keep checking modules. If it fails, record an optional failure and keep checking.
    - ***sufficient***: If this module is successful and no previous modules marked as required have failed, skip the rest of the modules. If this module fails continue checking.
- ***module_path***: Path to the PAM library for this module. If the module is in the default PAM location, only the name of the module need be specified. If the module is not located in the default location, specify the full path to the module.
- ***module_options***: Specific options that are passed to the modules and affect the behavior of the module. The following options are of interest in this configuration:
    - ***try_first_pass***: If more than one authentication method is configured, try the first password entered on subsequent authentication modules; only prompt for a password again if the passwords differ.
    - ***use_first_pass***: If more than one authentication method is configured, use the first password entered on all subsequent authentication modules. Do not prompt for another password if the first one fails on subsequent modules.

Example Solaris 9 /etc/pam.conf file: Below is a Solaris 9 /etc/pam.conf that has been configured to use both the LDAP and Kerberos PAM modules. The entries in bold have been added to the original, default file. Since, in this configuration, we do not want to require that LDAP and/or Kerberos authentication succeed, all LDAP and Kerberos entries have been configured with either the sufficient or the optional control flag. The service modules are stackable: each service is tried in turn in the order in which they appear in the file. Therefore, the placement of Kerberos login authentication with the sufficient flag above UNIX or LDAP authentication is significant. For login, if the Kerberos authentication succeeds, system and LDAP authentications do not need to succeed.

/etc/pam.conf

```
login   auth requisite          pam_authtok_get.so.1
login   auth required          pam_dhkeys.so.1
login   auth required          pam_dial_auth.so.1
login  auth sufficient       pam_krb5.so.1   try_first_pass
login   auth required          pam_unix_auth.so.1
login  auth sufficient       pam_ldap.so.1 try_first_pass

rlogin  auth sufficient        pam_rhosts_auth.so.1
rlogin  auth requisite         pam_authtok_get.so.1
rlogin  auth required          pam_dhkeys.so.1
rlogin auth sufficient       pam_krb5.so.1 try_first_pass
rlogin  auth required          pam_unix_auth.so.1
rlogin auth optional         pam_ldap.so.1 try_first_pass

rsh     auth sufficient        pam_rhosts_auth.so.1
rsh     auth required          pam_unix_auth.so.1
ppp     auth requisite         pam_authtok_get.so.1
ppp     auth required          pam_dhkeys.so.1
ppp     auth required          pam_unix_auth.so.1
ppp     auth required          pam_dial_auth.so.1

# Default definitions for authentication management
# Used when service name is not explicitly mentioned for authentication
other   auth requisite          pam_authtok_get.so.1
other   auth required          pam_dhkeys.so.1
other  auth sufficient       pam_krb5.so.1  try_first_pass
other   auth required          pam_unix_auth.so.1
other  auth sufficient       pam_ldap.so.1 try_first_pass

passwd  auth required         pam_passwd_auth.so.1

cron    account required       pam_projects.so.1
cron    account required       pam_unix_account.so.1
cron   account optional     pam_krb5.so.1

other   account requisite       pam_roles.so.1
other   account required        pam_projects.so.1
other   account sufficient    /usr/lib/security/pam_krb5.so.1
other   account required        pam_unix_account.so.1

other   session required         pam_unix_session.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password required       pam_authtok_store.so.1
other  password sufficient    pam_krb5.so.1   try_first_pass
other  password sufficient       pam_ldap.so.1   use_authtok
```

In this configuration, LDAP authentication is configured, but not used as a primary authentication method. The main method of authentication in this example is Kerberos. The use of AD LDAP authentication for all purpose user authentication is not scalable, and Microsoft recommends against it. Instead Microsoft recommends that LDAP's primary role be to serve as the directory and identity store, with Kerberos providing authentication services in the enterprise.

Example Fedora 4 /etc/pam.d/system-auth file: Below is a sample Fedora 4 /etc/pam.d/system-auth file that has been configured to use both the LDAP and Kerberos PAM modules. The entries in bold have been added to the original, default file. This PAM file allows either Kerberos or LDAP authentication to be sufficient for successful authentication. Kerberos is the primary method of authentication configured in this example.

/etc/pam.d/system-auth

```
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_krb5.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so broken_shadow
account     sufficient    /lib/security/$ISA/pam_localuser.so
account     sufficient    /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
account     [default=bad success=ok user_unknown=ignore] /lib/security/$ISA/pam_krb5.so
account     [default=bad success=ok user_unknown=ignore] /lib/security/$ISA/pam_ldap.so
account     required      /lib/security/$ISA/pam_permit.so

password    requisite     /lib/security/$ISA/pam_cracklib.so retry=3
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok shadow
password    sufficient    /lib/security/$ISA/pam_krb5.so use_authtok
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_krb5.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

4. Infrastructure Preparation

4.1 DNS

Active Directory Kerberos requires that DNS be configured and must support the following functionality:

- The DNS solution must be standards-based. The solution can be either Microsoft Server 2000/2003 DNS or Bind DNS.
- Service locator records must be supported. This is required for W2000/2003 Active Directory. Service locator records are used to locate the domain controllers, global catalog servers, Kerberos servers, LDAP servers, and the KPASSWD servers.

Other desired capabilities that the DNS solution should provide include support of dynamic updates and support for incremental zone transfers. Active Directory integrated DNS provides both these desired capabilities.

4.1.1 DNS Service Locator Records Required by Active Directory

The configuration presented in this tech report utilizes Active Directory integrated DNS.

In this solution there are two domain controllers for the domain ellie.com: dell1550-15.ellie.com and athena.ellie.com. Both DCs are also designated as global catalog servers. Below are the service locator records configured by AD integrated DNS for this configuration. If a bind DNS solution is implemented, the service locator records must be manually configured.

DNS Service Locator Records for ellie.com		
_ldap_tcp.ellie.com	SRV 0 100 389	athena.ellie.com
_kerberos_tcp.ellie.com	SRV 0 100 88	athena.ellie.com
_kpasswd_tcp.ellie.com	SRV 0 100 464	athena.ellie.com
_kerberos_udp.ellie.com	SRV 0 100 88	athena.ellie.com
_kpasswd_udp.ellie.com	SRV 0 100 464	athena.ellie.com
_gc_tcp.ellie.com	SRV 0 100 3268	athena.ellie.com
_ldap._tcp.dc._msdcs.ellie.com	SRV 0 100 389	athena.ellie.com
_ldap_tcp.ellie.com	SRV 0 100 389	dell1550-15.ellie.com
_kerberos_tcp.ellie.com	SRV 0 100 88	dell1550-15.ellie.com
_kpasswd_tcp.ellie.com	SRV 0 100 464	dell1550-15.ellie.com
_kerberos_udp.ellie.com	SRV 0 100 88	dell1550-15.ellie.com
_kpasswd_udp.ellie.com	SRV 0 100 464	dell1550-15.ellie.com
_gc_tcp.ellie.com	SRV 0 100 3268	dell1550-15.ellie.com
_ldap._tcp.dc._msdcs.ellie.com	SRV 0 100 389	dell1550-15.ellie.com

Table 1) DNS service locator records used by Active Directory.

Note: See the Microsoft KB Article “Integrating Windows 2000 DNS into an Existing BIND or Windows NT 4.0-Based DNS Namespace” for more information on using bind DNS in an Active Directory environment: http://support.microsoft.com/kb/255913.

4.2 NTP Time Synchronization

Kerberos 5 authentication is dependent upon the synchronization of time between the clients and the Kerberos KDCs. By default, if the time of the client and the time of the KDC do not match within 5 minutes, Kerberos authentication will fail. All Active Directory based Windows authentication will use Kerberos by default. As well, any NetApp storage system that is configured to join an Active Directory domain will use Kerberos for authentication. In this configuration, with UNIX or Linux clients using Kerberos authentication, the UNIX clients must be synchronized to a common time source as well.

Therefore; in a Kerberos environment, it is essential that time services be configured to ensure that time is synchronized across the enterprise. Any standards-based implementation can be used.

This solution utilizes W2003 based time services. The W2003 solution is based on SNTP.

The primary, master time server in a Windows 2003 environment is the domain controller designated as the PDC emulator. The PDC emulator in ellie.com is athena.ellie.com. All member servers and Windows clients in the domain will synchronize with the domain controllers without any extra configuration by the Windows administrator. However, UNIX or Linux hosts must be configured to synchronize to the Windows domain controllers or to a time service, which is in turn authoritative to the Windows environment (such as an Internet time server or an internal time server, which is designated authoritative for the entire enterprise).

The Windows master time server needs, in turn, to be synchronized with a reliable time source, either to an internal source or to an Internet time server.

This solution uses an Internet time server.
A list of publicly available time servers can be found at www.ntp.org. Many of them do not have restrictions, but do require e-mail notification.

4.2.1 Configuration of a W2003 PDC Emulator to Synchronize with an Internet Time Source

The Windows 2003 time service is configured with w32tm, a command line tool included with the standard Windows installation.

The following three steps set up and activate time synchronization with an Internet time source:

1. **w32tm /config /syncfromflags:manual /manualpeerlist:Peerlist**
   PeerList is a comma-separated list of DNS names or IP addresses of the desired Internet time sources.
2. **w32tm /config /reliable:YES**
   This command configures the Windows time service to announce itself as a reliable time source so other computers can synchronize to it.
3. **w32tm /config /update**
   This command notifies the time service of the changes to the configuration, causing the changes to take effect.

Below are the commands run for this configuration to set up time synchronization to an Internet time source:

1. **w32tm /config /syncfromflags:manual /manualpeerlist:t2.timegps.net,ntp1.sf-bay.com**
2. **w32tm /config /reliable:YES**
3. **w32tm /config /update**

4.2.2 NTP Configuration on Solaris 9

To configure time services on Solaris, perform the following three steps:

1. Copy the template file provided to ntp.conf:

   cp /etc/inet/ntp.client ./ntp.conf

2. Modify ntp.conf to include the time server that will be used by this client.

   Minimal required entries in ntp.conf include the time servers that the client should synchronize with and the location of the drift file, which is used to record information regarding the accuracy of the local clock.

/etc/inet/ntp.conf

```
server  athena.ellie.com
server  dell1550-15.ellie.com
driftfile /etc/ntp.drift
```

3. The ntpd daemon must be restarted in order for configuration changes to take effect.

   /etc/init.d/xntpd stop
   /etc/init.d/xntpd start

4.2.3 NTP Configuration on Linux

To configure time services on the Linux clients, perform the following two steps:

1. Modify ntp.conf to include the time servers that will be used by this client.

   Minimal required entries in ntp.conf include the time servers that the client should synchronize with.

/etc/ntp.conf

```
# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1

server 10.32.16.60
server 172.17.36.112
```

2. For the configuration changes to take effect, restart the ntpd daemon:

   /etc/init.d/ntpd restart

4.3 Network Infrastructure

Ensure that the infrastructure supports reliable communication between clients, the NetApp storage system, DNS servers, time servers, and Active Directory domain controllers.

Recommendations:

- To ensure that clients can find the Active Directory LDAP and Kerberos servers there must be reliable network connectivity between the clients and DNS servers containing the LDAP and Kerberos service records. If possible this should be a high-bandwidth connection.
- Clients must have reliable connections to domain controllers which host both the LDAP and Kerberos services. If possible this should be a high-bandwidth connection.

- When the enterprise contains more than one domain and/or utilizes universal groups, there must be adequate connectivity from domain controllers to a global catalog server. If possible, this should be a high-bandwidth connection.
- If the enterprise is located in multiple locations, with the locations connected with less than high-bandwidth connections, it is recommended that Active Directory sites be configured. Sites are a way to group resources together that are all within a local high-bandwidth zone. Domain controllers within a site are preferred over domain controllers within another site, which enhances responsiveness of domain controllers to client and NetApp storage requests. Additionally, there should be at least one global catalog server within every site.
- If clients from other domains access resources on the NetApp storage system, ensure that there is reliable connectivity between the storage system and all domain controllers with users who access resources on the storage system.

  Why? If multiprotocol is enabled and a user accesses data on a NetApp storage device through CIFS, the user is always mapped to the user's corresponding UNIX identity. The initial access request via CIFS contains a user's SID and all the SIDs of the groups to which the user belongs. However, the request does not contain the user's login name. In order to map a Windows user to a UNIX user the NetApp storage device must retrieve the Windows login name. This necessitates a query to the user's domain controller.

  Conversely, if a UNIX user accesses data in a volume that is either mixed or NTFS security style, again, a user mapping is performed. The storage system must contact the user's domain controller to obtain the user's Windows login name.

  The user mapping information is cached, by default, for 20 minutes. If user information changes infrequently and network connectivity might be an issue, the user mapping cache time can be increased.

  The option wafl.wcc_minute_valid is used to increase the cache time:

  options wafl.wcc_minutes_valid *number_of_minutes*
  Specifies the number of minutes a WAFL® credential cache entry is valid. The value can range from 1 through 20160. The default is 20.

5. NFS Kerberos Security on the NetApp Storage System

5.1 Kerberized NFS on the NetApp Storage System

NetApp storage systems support Kerberos V5 security on NFS exports in Data ONTAP 6.4 and above. Kerberos security for NFS is configured with "nfs setup." After Kerberos for NFS is configured, specific exports are configured to use Kerberos security by creating the appropriate entries in the exports file. The following security modes are supported and are configurable in the /etc/exports file:

sec=secflavor[:secflavor]...
Allow access to the mounted directory only using the listed security flavors. If no sec directive is provided, then the default of sys is applied to the export. The sec directive may appear multiple times in a rule, which each appearance setting the context of the following directives: anon, nosuid, ro, root, and rw. The contexts apply in order. If only one security context is provided in an export, then it applies regardless of where it appears in the export. Note that any given security flavor can only appear once in an export rule.

The supported security flavors are:

- **sys**
  for UNIX style security based on uids and gids
- **krb5**
  for Kerberos Version 5 authentication.
- **krb5i**
  for Kerberos Version 5 integrity service
- **krb5p**
  for Kerberos Version 5 privacy service

The Kerberos authentication service verifies the identity of the users accessing the storage system on all accesses, and also verifies to the client that the responses are from the storage system. The integrity service provides a strong assurance that the messages have not been tampered with. The privacy service ensures that messages intercepted on the wire cannot be read by any other party. The integrity and privacy services both include authentication. The default security flavor is sys.

The security flavor of none can also be applied to an export. If the client uses this flavor, then all requests get the effective UID of the anonymous user. Also, if a request arrives with a security context that is not present in the export, and none is allowed, then that request is treated as if it arrived with the flavor of none.

Examples:

- **/vol/vol5 -ro=.farm.mycompany.com,sec=krb5,rw,anon=0**

  If the security flavor is sys, then all hosts in the DNS subdomain of farm.mycompany.com are granted ro access. If the security flavor is krb5, then all hosts are granted rw access.

- **/vol/vol6 -sec=sys:none,rw,sec=krb5:krb5i:k4b5p,rw,anon=0**

  If the security flavor is sys or none, then all hosts are granted rw access, but effectively all root access is blocked. If the security flavor is from one of the secure krb5, krb5i, or krb5p, then rw and effectively root access are both granted.

5.1.1 KDC Realms and the Storage System

Currently Data ONTAP supports one and only one KDC realm per physical storage system. If the storage system is configured during CIFS setup to participate in an Active Directory domain, the AD KDCs are configured as the sole KDC realm for the physical storage system. With this configuration, if the customer desires to use Kerberized NFS mounts, the Active Directory KDCs must be used for NFS Kerberos security. NFS setup will not allow a UNIX based KDC to be configured if the storage system is joined to an AD domain from CIFS setup.

If the storage system is configured to be part on a Windows NT® domain or is configured as part of a local Windows workgroup (option 2, 3, or 4 in CIFS setup), then a UNIX based KDC may be configured through NFS setup.

The one exception to one KDC per physical storage system is if MultiStore® is licensed on the system and vFiler™ appliances are configured. Each vFiler instance can be configured with its own KDC realm. However, within the vFiler instance, either an AD KDC is configured OR a UNIX KDC is configured.

Current Data ONTAP implementations:

- Support DES encryption (triple DES encryption not supported).
- Do not support cross-realm authentication.

5.2 Prior to Enabling Kerberos for NFS on the Storage System: Infrastructure Preparation

Prior to configuring and enabling Kerberos for NFS on the storage system, several parameters on which Kerberos depends must be configured. As discussed previously, Active Directory Kerberos requires proper DNS configuration and proper time synchronization. Before the storage system can participate in an Active Directory domain environment CIFS setup must be run and the storage system must be joined to a domain. When using Active Directory Kerberos for NFS, NFS setup uses the information that was previously configured during "CIFS setup" to gather information on the Kerberos KDCs.

This configuration uses Active Directory LDAP as a directory store. The nsswitch.conf file reflects this. Nsswitch.conf name service maps may vary in the customer environment depending on what name services are in use.

For more information on the use of Active Directory LDAP as a unified directory store, please see TR 3458, "Unified Windows and UNIX Authorization Using Microsoft Active Directory LDAP as a Directory Store."

Infrastructure Preparation Tasks	
/etc/resolv.conf	Must contain name server addresses for DNS servers which contain the appropriate Active Directory service locator records.
/etc/nsswitch.conf	Ensure that DNS is configured for the hosts map.
DNS options	Ensure that dns is enabled and that the correct dns domain name is configured.
CIFS Setup	Completed prior to enabling LDAP.
Time Server Setup	Active Directory Kerberos requires that the storage system's time match domain controller time within five minutes.

Table 2) Infrastructure preparation tasks.

Below are the storage system options and configuration files used on the storage system (Data ONTAP 7.1P2) for this configuration:

- **/etc/resolv.conf**

```
nameserver 10.32.16.60
nameserver 172.17.36.112
```

- **/etc/nsswitch.conf**

```
hosts:      files  dns    nis
passwd:     files  ldap
netgroup:   files  nis
group:      files  ldap
shadow:     files  ldap
```

Note: It is recommended that files always be the first entry for all maps. It allows quick resolution of minimally required entries even if network connectivity is lost.

- **options dns**

```
dns.cache.enable            on
dns.domainname              ellie.com
dns.enable                  on
```

- **cifs domaininfo**

```
NetBios Domain:                 ELLIE
Windows 2000 Domain Name:       ellie.com
Type:                           Windows 2000
Filer AD Site:                  java1
(output truncated)
```

- **options timed**

```
timed.enable                    on
timed.proto                     ntp
timed.servers                   10.35.8.60
```

5.3 Enable Kerberos Security for NFS Exports on the NetApp Storage System

Kerberos security for NFS is configured and enabled on the NetApp storage system through the "**NFS Setup**" command.

When using Active Directory Kerberos for UNIX NFS security, if the storage system is already joined to an Active Directory domain, configuration is straightforward. The storage system will automatically select the AD domain used in CIFS setup for use with NFS Kerberos. The storage system will take much of the configuration from CIFS setup and incorporate it into the NFS setup:

1. Run "nfs setup" and turn the option "nfs.kerberos.enable" on to enable NFS security on the storage system:

```
options nfs.kerberos.enable on
nfs setup
Enable Kerberos for NFS? y
The filer supports these types of Kerberos Key Distribution Centers (KDCs):

    1 - UNIX KDC
    2 - Microsoft Active Directory KDC

Enter the type of your KDC (1-2): 2
Kerberos now enabled for NFS.
NFS setup complete.
```

2. Export volumes or qtrees with Kerberos security:

   Modify /etc/exports file to include the desired Kerberos security option. Supported options are krb5,krb5i, and krb5p. For configuration changes to take effect, reexport the exports file with the command "**exportfs –r**."

   Example entry in /etc/exports:

```
/vol/vol2      -sec=sys:krb5,rw,root=10.0.1.1
```

6. Configure Solaris 9 to Use Active Directory Kerberos for Authentication

This section will outline the procedure used to configure a Solaris 9 client to use Active Directory Kerberos for authentication.

Solaris 9's Kerberos implementation is integrated into the Sun™ SEAM (Sun Enterprise Authentication System) architecture. Because of customization, SEAM enabled clients are not able to natively use Active Directory-based Kerberos for authentication. Therefore, MIT Kerberos must be downloaded and installed on a Solaris 9 client before it can be configured to use AD Kerberos.

MIT Kerberos versions prior to 1.3.1 have known issues when using Microsoft Active Directory Kerberos. Versions prior to 1.3.1 have limitations on the UDP packet size that results in the truncation of group lists in the authentication reply packets. Earlier versions of MIT Kerberos also have security issues. When downloading MIT Kerberos for install, newer versions are recommended.

Prior to building and installing MIT Kerberos from source, several supporting GNU packages are needed for the build and install process to succeed. Solaris 9 does not have these packages installed by default. See Appendix B for a list of needed supporting packages, where to download them, and the methodology for installing the supporting packages.

6.1 Infrastructure Preparation

Prior to installing MIT Kerberos on the Solaris 9 client, the DNS file, /etc/resolv.conf, and time services must be properly configured to ensure proper Kerberos client functionality.

Unlike the native Windows Active Directory environment that requires that DNS service locator records be used when locating Kerberos KDCs, the UNIX clients do not depend on DNS for locating Kerberos services. Instead UNIX clients refer to entries in the Kerberos configuration file, /etc/krb5/krb5.conf, to locate the KDCs. However, for proper network connectivity, the UNIX client must be able to resolve host names, with DNS being the most universally deployed method.

Time synchronization between the Kerberos KDCs and the clients is critical in a Kerberos environment. If the times differ more than five minutes (default), authentication will fail. Configure time services on the client with the /etc/ntp.conf file.

Additionally, the order of UNIX name services is configured with the nsswitch.conf file. DNS must be added to the hosts map before the NSS will use DNS for host name resolution.

This configuration uses Active Directory LDAP as a directory store. The nsswitch.conf file reflects this. Nsswitch.conf name service maps may vary in the customer environment depending on what name services are in use.

For more information on the use of Active Directory LDAP as a unified directory store, please see TR 3458, "Unified Windows and UNIX Authorization Using Microsoft Active Directory LDAP as a Directory Store."

Infrastructure Preparation Tasks	
/etc/resolv.conf	Must contain name server addresses for DNS servers to be used for host name resolution. For this configuration, the Linux clients are using the same AD integrated DNS servers as are used by the Windows environment.
/etc/nsswitch.conf	Add the DNS name service to the host map.
Time Server setup	Active Directory Kerberos requires that the storage system time match domain controller time within five minutes.

Table 3) Infrastructure preparation tasks.

Below are the configuration files used on the Solaris 9 client for this configuration:

- **/etc/resolv.conf**

```
nameserver 10.32.16.60
nameserver 172.17.36.112
search ellie.com
```

- **/etc/nsswitch.conf**

```
host:           files dns nis
passwd:         files ldap
shadow:         files ldap
group:          files ldap
```

- **/etc/inet/ntp.conf**

```
server  athena.ellie.com
server  dell1550-15.ellie.com
driftfile /etc/ntp.drift
```

NSCD and PAM

The name service cache daemon is a go-between for applications that need name services and the mechanisms providing the name services. NSCD is installed by default on Solaris 9. NSCD provides the method for retrieving information needed during authentication, but the NSCD does not play a direct role in authentication on PAM-enabled systems. PAM authentication occurs completely separate from NSS requests.

6.2 Configuring Kerberos on the Solaris 9 Client

6.2.1 Build and Install MIT Kerberos on Solaris 9

1. Verify that all the prerequisite GNU tools are installed.
2. Ensure that your PATH includes /usr/local/bin:/usr/ccs/bin.
3. Download the latest version of MIT Kerberos from: http://web.mit.edu/kerberos/www/.
4. Extract the source files:

   ```
   tar xvf  krb5-1.4.2-signed.tar
   gunzip krb5-1.4.2.tar.gz
   tar xvf krb5-1.4.2.tar
   ```

5. Change directories into./krb5-1.4.2/src.
6. Configure and compile the source code:

   ```
   ./configure
   make
   make install
   ```

7. Create links to provide the correct location for MIT Kerberos.

   The install places the Kerberos files into the default directories, which are already included in the PATH previously configured. However, MIT Kerberos expects two of its configuration files to be in the /etc directory. The install places these files in the /etc/krb5 directory. Creating the links below allows Kerberos to find configured information at the expected location.

```
ln –s /etc/krb5/krb5.conf /etc/krb5.conf
touch /etc/krb5/krb5.keytab
ln –s /etc/krb5/krb5.keytab /etc/krb5.keytab
```

Note: The /etc/krb5/krb5.keytab file does not exist by default. The file will be populated later when keytab files that we will create on the Active Directory KDC server are imported into the UNIX client's /etc/krb5/krb5.keytab file.

8. To access MIT Kerberos man pages, modify the MANPATH:

```
MANPATH=/usr/local/man:/usr/share/man:/usr/man:$MANPATH; export MANPATH
```

6.2.2 Configuring Kerberos with the /etc/krb5/krb5.conf File

The Solaris 9 Kerberos configuration file is /etc/krb5/krb5.conf. (The Solaris 9 default location is /etc/krb5/krb5.conf; however, MIT Kerberos expects the file to be in the /etc/ directory. Therefore, in the previous section we created a link to the default location.)

Krb5.conf contains all the realm information and default settings including:

- The location of the KDCs
- The location of the KDC admin server
- Host name mappings
- Default settings

The krb5.conf file is made up of stanzas. The file can be made up of any or all of the stanzas. Below is the list of stanzas available and the entries used in our configuration:

/etc/krb5/krb5.conf	
[libdefaults]	This contains the default values used by the Kerberos 5 library.
default_realm:	The default realm for the client
dns_lookup_kdc:	Indicates if DNS SRV records should be used to locate KDCs and other servers for a realm. This value will be false when using Active Directory Kerberos.
dns_lookup_realm:	Indicates if DNS TXT records should be used to locate KDCs and other servers for a realm. This value will be false when using Active Directory Kerberos.
default_tgs_enctypes:	The list of supported encryption types on the KDC. Microsoft Kerberos supports only the MIT Kerberos encryption types "**des-cbc-crc**" and "**des-cbc-md5**."
default_tkt_enctypes:	The list of encryption types that should be requested by the client. Microsoft Kerberos supports only the MIT Kerberos encryption

		types “**des-cbc-crc**” and “**des-cbc-md5.**”
[realms]		Contains subsections for each Kerberos realm describing realm-specific information.
	kdc:	The host name or IP address of a computer running KDC for the realm. Port number is optional. More than one kdc can be specified.
	admin_server:	The host name or IP address of a computer running a Kerberos administration server. Only one admin_server can be specified.
	kpasswd_protocol	Used when Active Directory is used as the Kerberos realm. Optional setting.
[domain_realm]		Contains information regarding the mapping of domain and subdomain names to Kerberos realms. Contains a list of domain names to which the Kerberos realm maps. Subdomains (indicated by preceding period) need to be listed before parent domains.
	Example:	**[domain_realm]** **.mit.edu = ATHENA.MIT.EDU** **mit.edu = ATHENA.MIT.EDU** **.media.mit.edu = MEDIA-LAB.MIT.EDU** **media.mit.edu = MEDIA-LAB.MIT.EDU** **.ucsc.edu = CATS.UCSC.EDU**
[logging]		Contains details of the logging to be performed by Kerberos applications.
	default:	The logging method to be used in the absence of any other specific information.
[appdefaults]		Contains the default values used by any Kerberos 5 application.

[login]	Contains the default values used by the Kerberos 5 login program.
[capaths]	Contains details of cross-realm authentication paths for direct authentication.

Table 4) Krb5.conf stanzas and entries used when using Active Directory Kerberos.

For full documentation on Kerberos configuration files and Kerberos commands, see the MIT Kerberos documentation at **http://web.mit.edu/kerberos/www/krb5-1.4.**

6.2.3 Solaris 9 /etc/krb5/krb5.conf File

Below is the output from /etc/krb5/krb5.conf file used in this configuration.

```
/etc/krb5/krb5.conf

[libdefaults]
      default_realm = ELLIE.COM
      dns_lookup_realm = false
      dns_lookup_kdc = false
      default_tkt_enctypes =  des-cbc-crc ; or des-cbc-md5
      default_tgs_enctypes =  des-cbc-crc ; or des-cbc-md5

[realms]
      ELLIE.COM = {
                  kdc = 10.32.16.60:88
                  kdc = 172.17.36.112:88
                  admin_server = 10.32.16.60:749
                  kpasswd_protocol = SET_CHANGE
      }

[domain_realm]
      .ellie.com = ELLIE.COM
      ellie.com = ELLIE.COM

[logging]
      default = FILE:/var/krb5/kdc.log

[appdefaults]
      kinit = {
            renewable = true
            forwardable= true
      }
```

6.2.4 Creating and Installing Keytab Files

Kerberos realms are a collection of principals that are all stored in a common database. All users and computers that participate in the Kerberos realm must have a principal account in the Kerberos database. Additionally, every service offered by a host that will use Kerberos authentication must have a special service principal name. A service principal name (SPN) is the name by which a client uniquely identifies an instance of a service. Kerberos can use an SPN to authenticate a service. When a client wants to connect to

a service, it locates an instance of the service, composes an SPN for that instance, connects to the service, and presents the SPN for the service to authenticate. If a service does not have an SPN, Kerberos authentication to this service will fail.

Service principal names are multipart and take the form "*service\host_name@REALM_NAME*."

Windows Server account names are not multipart; therefore, it is not possible to directly create an account in Active Directory with the SPN format. We cannot directly create an account with the name "host/saturn.ellie.com@ELLIE.COM." To overcome this problem, the service principal name is created by mapping an SPN to a traditional Windows account.

Kerberos based on Microsoft stores SPNs in Active Directory. Hosts based on UNIX utilize a keytab file, called krb5.keytab. The keytab file is an encrypted, local, on-disk file that contains copies of SPN's keys. By using a keytab file, hosts based on UNIX which are configured to use AD Kerberos can request access to Kerberized services without supplying the SPN password. Typically, the Solaris keytab file will contain a key for the client's host service and a special "root" SPN that allows Kerberized NFS mounts to be done without specifically requesting root credentials. MIT Kerberos-based services (noninteractive) use the keytab to log on and use Kerberos services.

When using a KDC based on UNIX, the keytab file can be generated by the Kerberos admin utility. To populate the UNIX keytab file when using an Active Directory KDC, a keytab file must be created on the Active Directory server. The Microsoft keytab output file is used when merging the host and root keytabs into Solaris host's /etc/krb5/krb5.keytab file.

Two Microsoft command line tools are used to create the service principal names and to create keytab files for the SPNs.

- **setspn – used to create service principal names**
- **ktpass – used to create keytab files**

With the Solaris 9 configuration used in this report, we need to create two Microsoft SPN keytab files and import them into the Solaris 9 /etc/krb5/krb5.keytab file.

- **host – used for network services such as rlogin and telnet**
- **root – used for mounting nfs mounts**

6.2.4.1 Create and Import Keytab Files for Host and Root SPN

1) Create the host and root SPN.

Create two regular Windows users:

- You cannot map multiple service instances to the same user account.
- Create a separate user for each SPN needed on each host.
- When creating the user account clear the check box "User must change password at next logon."

```
saturn      -- used for the host SPN
saturnroot -- used for the root SPN
```

2) Create the SPN.

During SPN creation, the SPN is mapped to the Windows accounts created in Step 1.
Start a command line shell on the domain controller and enter the following commands:

```
setspn –A host/saturn.ellie.com saturn
setspn –A root/saturn.ellie.com saturnroot
```

Example:

```
setspn -A host/saturn.ellie.com saturn
Registering ServicePrincipalNames for
CN=saturn,OU=java_users,DC=ellie,DC=com
      host/saturn.ellie.com
Updated object
```

3) Verify SPN.

```
setSPN -L saturn
Registered ServicePrincipalNames for
CN=saturn,OU=java_users,DC=ellie,DC=com:
   host/saturn.ellie.com
```

4) Create the host and root keytab file.
The Windows command line utility, ktpass, is used to create the keytab files that are then securely transferred to the corresponding UNIX or Linux host.

The keytab is output to a file that you designate in the command line. The file name can be anything, but should be descriptive. The keytab files are later transferred to specific UNIX clients. A descriptive name ensures that the keytabs are not transferred to the incorrect host.

Example:

```
ktpass –princ host/saturn.ellie.com@ELLIE.COM –mapuser saturn –pass
blahblahpass –out saturnhost.keytab

ktpass –princ root/saturn.ellie.com@ELLIE.COM –mapuser saturnroot –pass
blahblahpass –out saturnroot.keytab
```

5) Transfer the keytab files to the UNIX clients.
Each host keytab file is created for a specific host. Ensure that each keytab is securely transferred to the correct host and that the files have UNIX permission rw-------.

6) Install the keytab files on the Solaris host using ktutil.
The Microsoft keytab files are merged with the UNIX client's /etc/krb5/krb5.keytab file with ktutil. The ktutil command invokes a subshell from which an administrator can read, write, or edit entries in a Kerberos V5 keytab file.

With ktutil we can read a keytab into memory, list the keytabs that are in memory, and write the keytabs in memory to the krb5.keytab file.

We can also delete keytab entries that are in memory. Deleting an entry with ktutil deletes it from memory only. It does not delete an entry in the keytab file. There is nothing in this utility that can be used to delete entries from a keytab file. If you perform a "write to keytab" action, it appends the keytab entries that are in memory to the existing entries in the specified file. It does not overwrite it. To eliminate keytabs from the krb5.keytab file, the file must be renamed and a new krb5.keytab created by importing the appropriate Microsoft keytab files. (Use this technique if the versions of the keytab files are outdated. See Section 6.4.2.2 for more information on outdated keytab files.)

For ease of use, place the Microsoft keytab files into the Solaris client's /etc/krb5/ directory.

Example:
Invoke the ktutil command:

```
ktutil
ktutil:  rkt saturn.keytab
```

```
ktutil:  rkt saturnroot.keytab
ktutil:  list
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
1   26      host/saturn.ellie.com@ELLIE.COM
2   10      root/saturn.ellie.com@ELLIE.COM
ktutil:  wkt /etc/krb5/krb5.keytab
ktutil:  q
```

Options:

rkt *keytab_file*	Reads the entries from the specified keytab file into memory.
wkt *keytab_file*	Writes the keytab entries in memory to specified file.
list	List the keytabs held in memory.
delent *slot_number*	Deletes an entry from the active ktutil keytab list.
q	Quit. Exit the utility.

7) Test of krb5.keytab validity.

```
kinit –k [–t /etc/krb5.keytab] host/saturn.ellie.com
```

-k indicates to use a keytab file.
-t is optional if using the default keytab file.

Example:

```
kinit –k  host/saturn.ellie.com
```

- The –t option is not used since the default keytab file is used.
- The REALM part of the SPN, @ELLIE.COM is not included in this example because it will be appended on to this request using the default realm configured in the krb5.conf file.
- No password is requested because the password is stored in the generated keytab file.

6.2.4.2 Mismatching Keytab Version Numbers

For correct keytab file functionality, versions of entries in the keytab file must match the version numbers of the SPN that is stored in Active Directory. Any changes to the user account to which the SPN is mapped will result in a change to the AD stored version number.

Mismatched version numbers will lead to the inability for users to log in to the client via Kerberos. It also causes problems with Kerberos mounted nfs file systems.

- Below are typical errors encountered if the versions in the keytab file don't match versions of SPN in AD.

  Example: Solaris 9 /var/adm/messages error messages.

  saturn login: [ID 537602 auth.error] PAM-KRB5 (auth): krb5_verify_init_creds failed: Key version number for principal in key table is incorrect
  saturn login: [ID 537602 auth.error] PAM-KRB5 (auth): krb5_verify_init_creds failed: Server not found in Kerberos database

- How to fix mismatched version numbers:

  1) Create a new keytab file on the Microsoft AD server using the method outlined above.

  2) Rename the /etc/krb5/krb5.keytab file.

3) Recreate the krb5.keytab file using the method above. If some of the SPN keytabs are still valid, use the original Microsoft keytab files for those SPNs and use the newly created keytab file for the SPNs that have mismatched versions.

- How to avoid mismatched keytab versions:
  - Do not make changes to the user to which the SPNs are mapped.
    - This includes changes to the password, changes to display name, etc.
    - Moving the user object to another location within AD also will cause a change to the Kerberos SPN version number.

6.3 Configuring PAM to Use the Kerberos PAM

The Kerberos PAM module allows UNIX and Linux clients to use Kerberos for authentication and account information. Kerberos authentication is the preferred method for centralized, secure client/server-based authentication. In this configuration, Active Directory LDAP is also configured through PAM as an authentication method. LDAP authentication has been configured for use during bind operations to the LDAP directory. LDAP binds allow control over what entities can connect to and perform operations on the LDAP directory. Microsoft recommends that LDAP authentication not be used as a primary method for authenticating users in the enterprise. LDAP authentication is not scalable in large environments and is suited for controlling access to the LDAP directory, but not as a primary method for user authentication.

To allow Kerberos to be used for login authentication edit /etc/pam.conf and add the following lines shown in bold to the stanzas shown below. In order to use Kerberos as the preferred, primary method for authentication, place the added lines in the indicated order within each section. This configuration will allow Kerberos authentication to be sufficient; UNIX system authentication and/or LDAP authentication do not have to succeed if Kerberos authentication succeeds. However, if Kerberos authentication fails, PAM will continue to check the authentication methods below for success. In this configuration, if Kerberos authentication fails, then UNIX system authentication must succeed (required flag) in order for login to succeed.

/etc/pam.conf

```
login   auth requisite          pam_authtok_get.so.1
login   auth required           pam_dhkeys.so.1
login   auth required           pam_dial_auth.so.1
login   auth sufficient         pam_krb5.so.1   try_first_pass
login   auth required           pam_unix_auth.so.1
login  auth sufficient          pam_ldap.so.1 try_first_pass

rlogin  auth sufficient         pam_rhosts_auth.so.1
rlogin  auth requisite          pam_authtok_get.so.1
rlogin  auth required           pam_dhkeys.so.1
rlogin  auth sufficient         pam_krb5.so.1 try_first_pass
rlogin  auth required           pam_unix_auth.so.1
rlogin auth sufficient          pam_ldap.so.1 try_first_pass

other   auth requisite          pam_authtok_get.so.1
other   auth required           pam_dhkeys.so.1
other   auth sufficient         pam_krb5.so.1  try_first_pass
other   auth required           pam_unix_auth.so.1
other  auth optional            pam_ldap.so.1 try_first_pass

other   account requisite       pam_roles.so.1
other   account required        pam_projects.so.1
other   account sufficient      /usr/lib/security/pam_krb5.so.1
other   account required        pam_unix_account.so.1
other  account optional         pam_ldap.so.1

other   session required        pam_unix_session.so.1
other   session optional        /usr/lib/security/pam_krb5.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password required       pam_authtok_store.so.1
other   password sufficient     pam_krb5.so.1   try_first_pass
other  password sufficient      pam_ldap.so.1   use_authtok
```

- This configuration is using Kerberos as the main method for user authentication. LDAP authentication is configured to allow successful LDAP binds, should anonymous binds to Active Directory LDAP be disallowed.
- Try_first_pass means that the password the user first enters will be tried for Kerberos authentication. If it is not valid, the user will be prompted for a Kerberos password.
- The actual pam.conf file contains additional sections that do not require modification. *Please see Section 3.5 for the output of a complete Solaris pam.conf file and more information on PAM.
- When using the above configuration, leave some user entries in the /etc/passwd and /etc/group files in order to provide the ability to log in to the client even if there is a problem with the Kerberos infrastructure. Additionally, unless it is company policy to require all logins be done through Kerberos authentication, do not make Kerberos a requirement. If it is required and network infrastructure prevents access to the KDCs, no user will be able to log in to a UNIX or Linux client.
- It is possible to make changes to the pam.conf file that inadvertently leaves you unable to access the client by any means. To rectify this, boot the Solaris client from CD, mount the root directory to a temporary mount, and modify the /etc/pam.conf file as needed.

6.4 Changes to the /etc/inetd.conf File

A number of Kerberized daemons are installed with MIT Kerberos 1.3.2. The Solaris 9 /etc/inetd.conf file must be modified to use the Kerberized daemons instead of the traditional unkerberized daemons.

For those services for which Kerberos login is desired, add lines that refer to the Kerberized daemon and comment out lines that refer to non-Kerberized daemons for all relevant services.

Example: Edited entries from the /etc/inetd.conf file:

```
#login   stream  tcp6   nowait  root    /usr/sbin/in.rlogind    in.rlogind
klogin   stream  tcp    nowait  root    /usr/local/sbin/klogind klogind –k -c

#ftp     stream  tcp6   nowait  root    /usr/sbin/in.ftpd       in.ftpd -a
ftp      stream  tcp    nowait  root    /usr/local/sbin/ftpd    ftpd –a

#telnet  stream  tcp6   nowait  root    /usr/sbin/in.telnetd    in.telnetd
telnet   stream  tcp    nowait  root    /usr/local/sbin/telnetd telnetd –a valid
```

6.5 Configuring NFS Kerberos Security on the Solaris Client

There is a two-step process to configure Kerberos security on the Solaris client:

1) Modify /etc/nfssec.conf file to enable krb5 options.

```
# Uncomment the following lines to use Kerberos V5 with NFS
#
krb5        390003  kerberos_v5    default -           # RPCSEC_GSS
krb5i       390004  kerberos_v5    default integrity     # RPCSEC_GSS
krb5p        390005  kerberos_v5    default privacy      # RPCSEC_GSS
```

2) Mount the storage system mount with krb5 mount options, either mount as root or put mount entries in the /etc/vfstab file.

```
mount  –o sec=krb5 eddie:/vol/vol2 /eddie_vol2
```

- The following option sets security mode for both NFSv3 and NFSv4 on a Solaris 9 client.

  sec=mode: Set the security flavor for this mount to "mode." The following four modes are supported.

  - **sec=sys**, the default setting, which uses local UNIX uids and gids to authenticate NFS operations (AUTH_SYS)
  - **sec=krb5**, which uses Kerberos V5 instead of local UNIX uids and gids to authenticate users
  - **sec=krb5i**, which uses Kerberos V5 for user authentication and performs integrity checking of NFS operations using secure checksums to prevent data tampering
  - **sec=krb5p**, which uses Kerberos V5 for user authentication and integrity checking and encrypts NFS traffic to prevent traffic sniffing (this is the most secure setting)

  Note: Local UNIX uids and gids can be stored in /etc/passwd & group files, NIS, or LDAP.

3) Troubleshooting Kerberized NFS mount problems:

- Some versions of Linux do not support all the krb5 modes. Check the nfs man pages on the client to determine what krb5 modes are supported.
- Ensure that the client has a host and a root keytab entry in the /etc/krb5.keytab file.
- For successful access to data in a Kerberized mount, ensure that the person accessing data has a credential in their cache (check with klist, if a user credential is not there, do a kinit to obtain a credential). See Section 7 for more information on manually obtaining Kerberos credentials.

7 Obtaining Kerberos Credentials

If PAM is configured to use Kerberos authentication as the primary method of authentication and the user has a Kerberos principal in the Kerberos database, credentials are obtained automatically during login.

If the user does not have a Kerberos principal or another method of authentication was successfully used without invoking Kerberos authentication, the user may need to manually obtain credentials before asking for access to services that require Kerberos authentication, such as a user who needs access to a Kerberized NFS mount.

The Kerberos utilities, klist, kinit, and kdestroy are used to list, obtain, or destroy Kerberos credentials.

Example: Manually Obtaining and Listing Kerberos Credentials

1) Check to see that the user does not have a credential in cache:

```
bash-2.05$ klist
klist: No credentials cache file found while setting cache flags(ticket cache /tmp/krb5cc_118)
```

2) Destroy the credential if it exists but is no longer valid:

```
bash-2.05$kdestroy
```

3) Request the credential:

```
bash-2.05$ kinit anne
Password for anne@ELLIE.COM:
```

4) Verify the credential:

```
bash-2.05$ klist
Ticket cache: /tmp/krb5cc_118
Default principal: anne@ELLIE.COM

Valid starting                Expires                Service principal
Wed Jan 04 19:21:24 2006  Thu Jan 05 05:21:24 2006  krbtgt/ELLIE.COM@ELLIE.COM
        renew until Wed Jan 11 19:21:24 2006
```

Example: Manually obtain an SPN credential using the keytab file:

To obtain the host or root SPN, do a kinit and reference the keytab file. If the default /etc/krb5.keytab file is used, the path to the keytab file does not need to be included.

```
bash-2.05$ kinit -k host/saturn.ellie.com
```

Example: An SPN is obtained and stored in cache when obtaining a TGS for services on a remote host:

Anne has a credential and /eddie_vol2 is already mounted with krb5 security style. Anne changes into the mounted directory. When she does so, a TGS ticket is requested for the storage system's NFS service. At this time the storage system's nfs SPN credential is cached in her credential cache.

```
bash-2.05$ cd /eddie_vol2
bash-2.05$ ls
dfm.exe.1                    pam.d_ftp
```

```
bash-2.05$ klist
Ticket cache: /tmp/krb5cc_118
Default principal: anne@ELLIE.COM

Valid starting     Expires            Service principal
Wed Jan 04 19:21:24 2006  Thu Jan 05 05:21:24 2006  krbtgt/ELLIE.COM@ELLIE.COM
        renew until Wed Jan 11 19:21:24 2006
Wed Jan 04 19:25:26 2006  Thu Jan 05 05:21:24 2006  nfs/eddie.ellie.com@ELLIE.COM
        renew until Wed Jan 11 19:21:24 2006
```

8 References

- Solution Guide for Windows Security and Directory Services for UNIX (updated 1/20/2006): www.microsoft.com/downloads/details.aspx?familyid=144F7B82-65CF-4105-B60C-44515299797D&displaylang=en&displaylang=en

  Note: The Microsoft Solution Guide has extensive information on Active Directory LDAP and Kerberos in a unified solution.

- Microsoft Kerberos Authentication Technical Reference: www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/4a1daa3e-b45c-44ea-a0b6-fe8910f92f28.mspx

- How the Kerberos Version 5 Authentication Protocol Works www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/4a1daa3e-b45c-44ea-a0b6-fe8910f92f28.mspx

- Solaris 9 System Administration Guide: Security Services http://docs.sun.com/app/docs/doc/817-0365

- For full documentation on Kerberos configuration files and Kerberos commands, see the MIT Kerberos documentation at http://web.mit.edu/kerberos/www/krb5-1.4

- DATA ONTAP 7.1 Command Reference http://now.netapp.com/NOW/knowledge/docs/ontap/rel71/html/ontap/cmdref/index.html

- DATA ONTAP 7.1 File Access and Protocols Management Guide http://now.netapp.com/NOW/knowledge/docs/ontap/rel71/html/ontap/filesag/index.htm

9. Appendix A: Configure Fedora 4 to Use Active Directory Kerberos for Authentication

This section will outline the procedure used to configure a Fedora 4 client to use Active Directory Kerberos for authentication.

Frequently, there are several Linux distributions and several versions of each distribution deployed within the customer environment. Most versions ship with a version of MIT-based Kerberos; however, versions prior to 1.3.1 have known issues when using Microsoft Active Directory Kerberos. Versions prior to 1.3.1 have limitations on the UDP packet size which result in the truncation of group lists in the authentication reply packets. Therefore, if the UNIX or Linux client is using an older MIT Kerberos implementation, a newer version must be downloaded and installed. Earlier versions of MIT Kerberos also have security issues. Newer versions are recommended.

The Fedora 4 client used in this configuration utilizes a newer version of MIT Kerberos that was used directly to configure authentication with Active Directory Kerberos. A separate build and install of MIT Kerberos was not needed.

If the customer would like the newest version of MIT Kerberos, see Section 6.2 for the procedure to download and install MIT Kerberos from source. See Appendix C for prerequisite GNU tools needed on Fedora 4 before installing MIT Kerberos from source.

9.1 Infrastructure Preparation

Prior to configuring and enabling Kerberos on the Fedora client, the /etc/nsswitch.conf file, the /etc/resolv.conf file, and time services must be properly configured to ensure proper Kerberos client functionality.

The Fedora 4 client's resolv.conf and nsswitch.conf files use the same configuration as the Solaris 9 client discussed in Section 6.1. Please see Section 6.1 for more details. The Fedora 4 client's ntp.conf file configuration differs from Solaris 9 and is shown below:

- **/etc/ntp.conf**

```
# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions
restrict 127.0.0.1
server 10.32.16.60
server 172.17.36.112
```

9.2 Configuring Kerberos on the Fedora 4 Client

Since the Fedora 4 client already has a version of MIT Kerberos that fully supports Active Directory Kerberos, a build and install of MIT Kerberos is not done. Instead, proceed directly to configuration of the Kerberos configuration file /etc/krb5.conf.

9.2.1 Configuring Kerberos with the /etc/krb5/krb5.conf File

The Fedora 4 Kerberos configuration file, /etc/krb5.conf, is used to configure all parameters needed by the system to access and use Kerberos.

The configuration of this file is identical to that of the Solaris 9 client. Please see Section 6.2.2 for an explanation of entries in this file and for output of the krb5.conf file used in this configuration.

9.2.2 Configuring PAM to Use the Kerberos PAM

The Kerberos PAM module allows UNIX and Linux clients to use Kerberos for authentication and account information. In this configuration, Active Directory LDAP is also configured through PAM as an authentication method. LDAP authentication has been configured for use during LDAP bind operations to the LDAP directory. LDAP binds allow control over what entities can connect to and perform operations on the LDAP directory. Microsoft recommends that LDAP authentication not be used as a primary method for authenticating users in the enterprise as LDAP user authentication is not scalable in large environments. Microsoft recommends the use of Kerberos for secure, scalable user authentication.

Fedora 4 has a separate directory, /etc/pam.d, which contains the PAM configuration files, one file for each configured service. For convenience, use the Linux configuration utility, authconfig, to configure both NSS and PAM services. The authconfig utility configures PAM services by modifying the centralized /etc/pam.d/system-auth file. Modification of the other files in the directory is not necessary for this configuration.

Please be advised, whenever the authconfig utility is run, manual changes to the system PAM file, /etc/pam.d/system-auth are overwritten. Other authentication and NSS configuration files are modified as well. For instance, if Kerberos is configured through authconfig, the Kerberos configuration file /etc/krb5.conf may be modified. If, LDAP name services or authentication is configured through this utility, the configuration file /etc/ldap.conf may be modified. Not all the changes are desired or appropriate when using Microsoft Active Directory for Kerberos authentication or AD LDAP as a directory store. Therefore, before running the authconfig utility, make backup copies of /etc/pam.d/system-auth, /etc/ldap.conf, and /etc/krb5.conf files. After running the utility, compare the new versions of these files with the saved backups. Modify the new files as necessary.

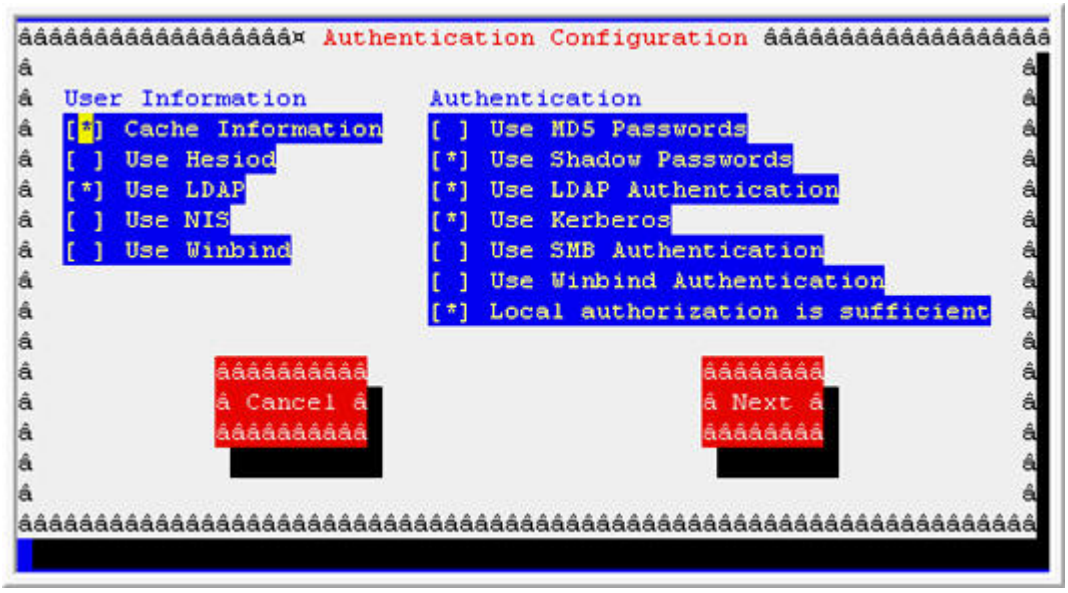
Follow the procedure below to configure PAM to use Kerberos for authentication and LDAP for name services and authentication:

1. Start the authconfig utility:

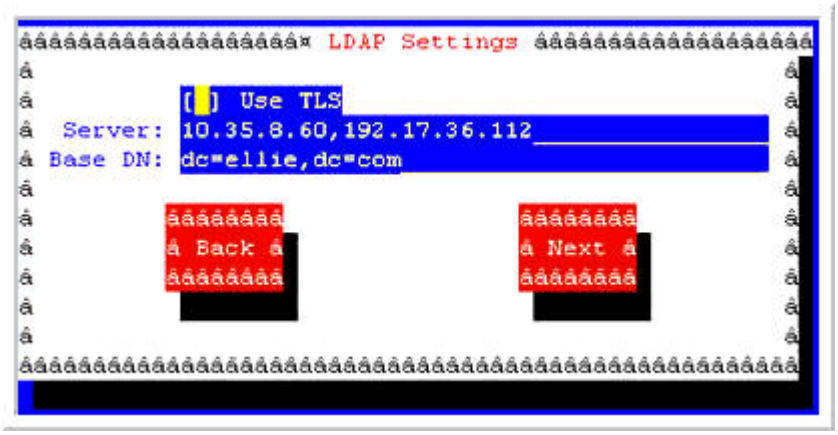
   persephone# authconfig

2. The "**Authentication Configuration**" screen is used to choose which name services this client should use and what authentication methods should be used. In this configuration both local /etc files and LDAP are used for name services.

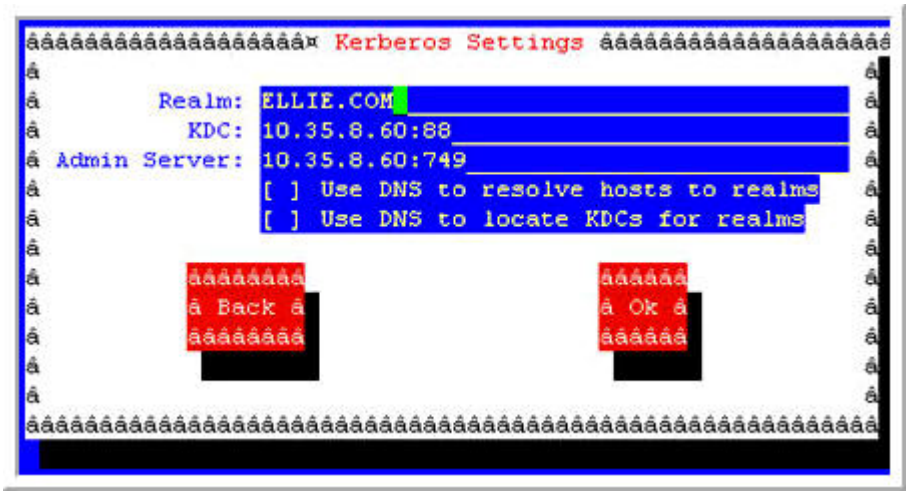
   System authentication, Kerberos authentication, and LDAP authentication are all configured authentication services. The option "**Local authentication is sufficient**" is chosen. This allows a local login for a user in the /etc/passwd file, but also exists in the Kerberos database or in the LDAP database, even if network service disruption causes LDAP or Kerberos servers to be unavailable.



3. The “**LDAP Settings**” screen is used to configure the LDAP settings. This configuration does not use TLS. LDAP communication will be over the default LDAP port, port 389.



4. The **“Kerberos Settings**” screen is used to configure Kerberos authentication. In this configuration, Kerberos is the main method deployed to authenticate users; however, both local system authentication and LDAP authentication are configured as well.



5. After clicking "**OK**," the /etc/pam.d/system-auth file has the following entries:

/etc/pam.d/system-auth

```
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_krb5.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so broken_shadow
account     sufficient    /lib/security/$ISA/pam_localuser.so
account     sufficient    /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
account     [default=bad success=ok user_unknown=ignore] /lib/security/$ISA/pam_ldap.so
account     [default=bad success=ok user_unknown=ignore] /lib/security/$ISA/pam_krb5.so
account     required      /lib/security/$ISA/pam_permit.so

password    requisite     /lib/security/$ISA/pam_cracklib.so retry=3
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok shadow
password    sufficient    /lib/security/$ISA/pam_krb5.so use_authtok
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_krb5.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

9.2.3 Creating and Installing Keytab Files

Similarly to the Solaris 9 configuration discussed in 6.2.3, keytab files must be prepared and installed on the Fedora 4 host; however, the type of SPN keytabs required differ. Host and root keytabs are not installed on the Fedora 4 client. Instead, an nfs SPN for the client is installed. Follow the procedure below to create the Fedora 4 client NFS SPN and keytab file and then to install the keytab on the Fedora 4 client.

1) Create the user to which the nfs SPN will be mapped.
Create a regular Windows user:

- Create a separate user for each nfs SPN needed on each Linux host.
- When creating the user account clear the check box "User must change password at next logon."

persephonenfs -- used for the nfs SPN

2) Create the SPN.
During SPN creation, the SPN is mapped to the Windows account created in Step 1.
On the domain controller, use the command line tool "setspn" to create the SPN:

setspn –A *service/hostname windows_account*

Example:

```
setspn -A nfs/persephone.ellie.com persephonenfs
Registering ServicePrincipalNames for
CN=persephonenfs,OU=java_users,DC=ellie,DC=com
      nfs/persephone.ellie.com
Updated object
```

3) Verify SPN.

```
setSPN -L persephonenfs
Registered ServicePrincipalNames for
CN=persephonenfs,OU=java_users,DC=ellie,DC=com:
   nfs/persephone.ellie.com
```

4) Create the nfs keytab file.
The Windows command line utility, ktpass, is used to create the keytab files that are then securely transferred to the corresponding UNIX or Linux host.

The command is executed on the domain controller, with the keytab being output to a file that you designate in the command line. The file name can be anything, but should be descriptive. The keytab files are later transferred to specific UNIX clients. A descriptive name ensures that the keytabs are not transferred to the incorrect host.

Example:

```
ktpass –princ nfs/persephone.ellie.com@ELLIE.COM –mapuser
persephonenfs –pass password –out persephonenfs.keytab
```

5) Transfer the keytab files to the Linux clients.
Each nfs keytab file is created for a specific host. Ensure that each keytab is securely transferred to the correct host and that the files have UNIX permission rw-------.

6) Install the keytab files on the Solaris host using ktutil.
The Microsoft keytab files are merged with the Linux client's /etc/krb5.keytab file with ktutil. The ktutil command invokes a subshell from which an administrator can read, write, or edit entries in a Kerberos V5 keytab file.

With ktutil we can read a keytab into memory, list the keytabs that are in memory, and write the keytabs in memory to the krb5.keytab file.

For ease of use, the Microsoft keytab file was placed into the client's /etc directory.

Example:
Invoke the ktutil command:

```
ktutil
ktutil:  rkt persephonenfs.keytab
ktutil:  list
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
  1    3      nfs/persephone.ellie.com@ELLIE.COM
ktutil:  wkt /etc/krb5.keytab
ktutil:  q
```

Options:

rkt *keytab_file*	Reads the entries from the specified keytab file into memory.
wkt *keytab_file*	Writes the keytab entries in memory to specified file.
list	List the keytabs held in memory.
delent *slot_number*	Deletes an entry from the ktutil keytab list held in memory.
q	Quit. Exit the utility.

7) **Test of krb5.keytab validity.**

kinit –k [–t /etc/krb5.keytab] *service|host_name@REALM*

-k indicates to use a keytab file.
-t is optional if using the default keytab file.

kinit –k nfs/persephone.ellie.com

- The –t option is not necessary as we are reading the keytab entry from the default file.
- The REALM part of the SPN, @ELLIE.COM is not included in this example because it will be appended on to this request using the default realm configured in the krb5.conf file.
- No password is requested because the password is stored in the generated keytab file.

9.2.4 Kerberized Daemons

A number of Kerberized daemons are installed with MIT Kerberos. Since the Fedora 4 client is configured, natively to use the MIT Kerberized daemons, no changes to the network services are needed.

9.3 Configuring NFS Kerberos Security on the Fedora 4 Client

Linux clients support Kerberized NFS mounts starting with 2.6. Kerberized mounts are supported on NFS versions 2, 3, and 4. To configure NFS mounts with Kerberos security follow the two-step process outlined below:

1. Configure the client to allow Kerberized NFS mounts:

   Fedora 4 uses the file "**/etc/sysconfig/nfs**" to enable Kerberized NFS mounts. Allow Kerberized NFS mounts by creating the "**/etc/sysconfig/nfs**" file with the following command:

   echo "SECURE_NFS=yes" > /etc/sysconfig/nfs

   - Failure to create the entry above in "/etc/sysconfig/nfs" file can lead to complaints about the rpc.gssd service such as the error below:

     root@persephone ~]# mount -t nfs4 -o sec=krb5 eddie:/vol/vol2 /mnt
     Warning: rpc.gssd appears not to be running.
     mount: Cannot allocate memory

2. Mount the storage system's exported volumes with krb5 mount options: either mount as root or put mount entries in the /etc/fsstab file.

   mount –o sec=krb5 eddie:/vol/vol2 /eddie_vol2

   - The following option sets security mode for NFSv2, NFSv3, and NFSv4 on a Fedora 4 client.

sec=mode: Set the security flavor for this mount to "mode." The following four modes are supported.

- **sec=sys**, the default setting, which uses local UNIX uids and gids to authenticate NFS operations (AUTH_SYS).
- **sec=krb5**, which uses Kerberos V5 instead of local UNIX uids and gids to authenticate users.
- **sec=krb5i**, which uses Kerberos V5 for user authentication and performs integrity checking of NFS operations using secure checksums to prevent data tampering.
- **sec=krb5p**, which uses Kerberos V5 for user authentication and integrity checking and encrypts NFS traffic to prevent traffic sniffing (this is the most secure setting).

3. Troubleshoot Kerberized NFS mount problems.

   - Some versions of Linux do not support all the krb5 modes. Check the nfs man pages on the client to determine what krb5 modes are supported on the client.
   - Ensure that the client has an nfs keytab in the /etc/krb5.keytab file.
   - Ensure that the gssd service is running.

     ```
     ps -eaf | grep gssd
     root     1674     1  0 May02 ?        00:00:03 rpc.gssd -m
     ```

     To start gssd:

     ```
     /etc/init.d/rpcgssd stop
     /etc/init.d/rpcgssd start
     ```

     - For successful access to data in a Kerberized mount, ensure that the person accessing data has a credential in their cache (check with klist, if a user credential is not there, do a kinit to obtain a credential).

9.4 Kerberos Client Tools

Use klist, kinit, and kdestroy to manage credentials on the Fedora 4 client. See Section 7 for more information on Kerberos client tools.

10. Appendix B: Solaris Prerequisite Packages Needed for MIT Kerberos Build and Install from Source

The following GNU packages are needed before MIT Kerberos can be built and installed from source.

Download the packages below from www.sunfreeware.com:

gcc-.3.4.2-sol9-sparc-local.gz
make-3.80-sol9-sparc-local.gz
autoconf-2.59-sol9-sparc-local.gz
automake-1.9-sol9-sparc-local.gz
m4-1.4.2-sol9-sparc-local.gz
libconv-1.8-sol9-sparc-local.gz
common-1.4.2-sunOS5.8-sparc-CSW.pkg.gz

Note: The prerequisite packages may themselves have other prerequisite packages. These will be noted on the sunfreeware download page.

10.1 Method for Prerequisite Package Install

Use the method below to install each of the prerequisite packages:

1. Change into the directory containing the downloaded packages.
2. Extract the Solaris package:

   gunzip *name_of_pkg.gz*

3. Install the package:

   pkgadd –d *name_of_pkg*

4. At the "**select packages(s) you wish to process (or 'all' to process all packages). (default: all [?,??,q]**" prompt, enter "**y**."
5. If the "**Do you want the directory created /usr/local [?,y,n,q]**" prompt is shown, enter "**y**."

10.2 Modify Environmental Variables

- The above utilities require Perl to be installed under /usr/local/bin, which is not the default Perl directory under Solaris 9. To enable the above utilities to use Perl, create a link.

  ln –s /usr/bin/perl /usr/local/bin/perl

- Additionally, before building and installing MIT Kerberos, the path must be modified to include /usr/ccs/bin and /usr/local/bin.

  PATH=/usr/ccs/bin:/usr/local/bin:$PATH; export PATH
  echo $PATH
  /usr/ccs/bin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/ucb:

10.3 Verify Package Installation

After installing the prerequisite packages, verify with the following command:

```
pkginfo | grep SMC
application SMCatk        atk
application SMCautoc      autoconf
application SMCautom      automake
application SMCexpat      expat
application SMCfontc      fontconfig
application SMCfreet      freetype
application SMCgcc342     gcc
application SMCglib       glib
application SMCgtk        gtk
application SMCiconv      libiconv
application SMCm4         m4
application SMCmake       make
application SMCpango      pango
application SMCzlib       zlib
```

11. Appendix C: Prerequisite Packages Needed for MIT Kerberos Build and Install on Fedora 4

1. Several supporting GNU packages are required when building and installing MIT Kerberos from source. If the developer packages are installed on the Linux client, the supporting GNU packages should already be installed. If they are not, install the missing rpms before proceeding.

   Validate that they are installed with the rpm command:

   rpm –qa | grep package_name
   gcc-3.2.3-20
   make-3.79.1-17
   automake-1.6.3-5
   autoconf-2.57-3
   m4-1.4.1-13

   If the rpms above are not installed perform either Step A or Step B below:

   - Step A: Install from the Red Hat Installation CDs with the rpm command:

     rpm –ivh /path_to_rpm/rpm_name.rpm

   - Step B: Use the up2date command to update to the latest version of these utilities.

2. Modify the PATH variable to include /usr/local/bin:

   echo $PATH
   /usr/local/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin

12. Revision History

Date	Name	Description
05/22/2006	Ellie Berriman	Creation