Windows Enterprise
OU Administrator Tips

Integrating RHEL5 Systems with Active Directory

John Dickerson and Beata Pruski
Created: December 4, 2009
Last Update: February 4, 2011

This document will show what steps[1] need to be taken to integrate a RHEL5 or RHEL6 system into the IASTATE Windows domain. In a nutshell, this document describes:

- How to set up Kerberos authentication on RHEL5 to use the Windows domain controllers.
- How to configure Samba for the IASTATE Windows domain.
- Joining a RHEL5 or RHEL6 system to the domain and creating trusted host credentials.
- Setting up Samba Winbind system to query the Active Directory for users and groups and provide login access controls.
- How to configure OpenLDAP to query against the Active Directory using Kerberos credentials.
- Using Kerberos authentication with SSH and CUPS printing.
- How to configure a client for secure NFS (NFSv4 with Kerberos) to use host and user credentials from the Windows domain.
- Some useful networking and security configurations that help with integration.

NOTES:

a. This document uses an example OU administrator username called "andyuser" and an example host name called "hydra.its.iastate.edu" (or simply "hydra"). Replace all the references to the example names with your OU administrator account or machine Fully Qualified Domain Name (FQDN) as appropriate.
b. This document assumes that your system has been registered with either the ISU Red Hat Network Proxy server or the Red Hat Network Satellite Server. This is required for installing the RHEL packages for proper integration.
c. A command that should be executed by root from a shell command line will be shown with the root prompt #. To type the 'hostname' command below, type only what comes after the #:

```
# hostname
```

[1] This document is based on:
- "Authenticate Linux Client with Active Directory" by Gil Kirkpatrick (http://technet.microsoft.com/en-us/magazine/2008.12.linux.aspx)
- "OpenSSH on Linux using Windows/Kerberos for Authentication" (http://port25.technet.com)

1. Configure Networking

Your system must be set up properly on the network. You'll need to have an on-campus IP address, either DHCP or statically assigned, and you will need to configure DNS and network time services to use campus servers.

a) Ensure that the DNS resolver for the Linux machine is set to use the same DNS name server that your Domain Controllers (DCs) use. For clients that use DHCP, you won't need to set this as the DHCP lease process will set the correct DNS configuration for you (including writing the contents of /etc/resolv.conf). Machines with statically assigned IP addresses should have the following lines in the file `/etc/resolv.conf`:

```
domain iastate.edu
; The search line below is optional. It sets the search domain to your
; your DNS subdomain.  Change as needed.
search engineering.iastate.edu
nameserver 129.186.142.200
nameserver 129.186.140.200
nameserver 129.186.1.200
```

b) We won't go into detail about configuring networking on RHEL, but for a host with a single network interface and a statically assigned IP address, it is usually sufficient to create the file `/etc/sysconfig/network-scripts/ifcfg-eth0` with the following lines:

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=129.186.145.124
NETMASK=255.255.255.0
ONBOOT=yes
```

c) Hosts with statically assigned IP addresses will need to set the host name and default gateway address. The file `/etc/sysconfig/network` should contain a HOSTNAME line that matches the DNS fully qualified hostname. It should also have a GATEWAY line for the default gateway address on your LAN:

```
HOSTNAME=hydra.its.iastate.edu
GATEWAY=129.186.145.254
```

d) For hosts with statically assigned IP addresses, it is a good idea to add the static IP address for the host to the file `/etc/hosts`. Add an entry below the localhost.localdomain entry like so:

```
129.186.145.124 hydra.its.iastate.edu hydra
```

e) You can reboot the machine to set the hostname and make the network active, or if you don't want to reboot, you can do:

```
# hostname hydra.its.iastate.edu
# service network restart
```

2. Configure Linux Time Synchronization

Kerberos authentication requires that the system clock on your Linux machine and the DCs' system clocks vary by no more than five minutes. You should configure your Linux system to use the Network Time Protocol (NTP) service on time.iastate.edu.

a) Make sure the ntpd package is installed:

```
# yum install -y ntpd
```

b) The file /etc/ntp.conf should contain the following lines:

```
server time.iastate.edu
restrict time.iastate.edu mask 255.255.255.255 nomodify notrap noquery
```

The first line above sets the time server to be time.iastate.edu. The second line sets some safe security options.

c) Use chkconfig to ensure that the ntpd daemon will start at next reboot. As root, type:

```
# chkconfig ntpd on
```

d) If you don't want to reboot yet, you can start the ntpd service manually:

```
# rdate -s time.iastate.edu
# service ntpd start
```

The **rdate** command above forces a one-time synchronization of the system clock to time.iastate.edu. This is important because ntpd will fail to start up if the current system time is too far out of sync.

3. Install Essential Packages

We need to ensure that all of the packages needed for Windows integration are installed, since not all are not installed by default.

The list of RHEL5 packages are:

samba-common*	Some base Samba components. [See the notes about Samba3x below.]
samba-client*	Samba client tools, including Winbind. [See the notes about Samba3x.]
krb5-workstation	Kerberos client utilities.
krb5-libs	Kerberos libraries.
openldap-clients	OpenLDAP client tools.
cyrus-sasl-gssapi	Provides GSSAPI (Kerberos) SASL support. Needed by OpenLDAP.
hesinfo	Client tool for querying Hesiod.
nfs4-acl-tools	Useful for reading/setting ACLs on NFSv4.

The list of RHEL6 packages are:

samba-common	Some base Samba components.
samba-winbind	Samba winbind.
samba-client	Samba client tools.
krb5-workstation	Kerberos client utilities.
krb5-libs	Kerberos libraries.
openldap-clients	OpenLDAP client tools.
cyrus-sasl-gssapi	Provides GSSAPI (Kerberos) SASL support. Needed by OpenLDAP.
hesinfo	Client tool for querying Hesiod.
nfs4-acl-tools	Useful for reading/setting ACLs on NFSv4.

To install all of these packages with one command, you can do:

```
# yum install –y samba-common samba-client krb5-workstation krb5-libs \
  openldap-clients cyrus-sasl-gssapi hesinfo nfs4-acl-tools
```

Samba3x

The default Samba packages in RHEL5 that start with "samba-" are based on Samba 3.0.33. However, in RHEL 5.6, Red Hat provides an alternate set of Samba packages based on Samba 3.5.4. (This is the same version of Samba used by default in RHEL 6). Samba 3.5.4 contains several enhancements that justify using them instead of the default Samba packages. In particular, Samba 3.5.4 and later support "LDAP SASL signing" used by the Windows domain controllers. Clients that don't use LDAP SASL signing cause a security event to appear in the domain controller logs when they query the Active Directory. Winbind 3.5.4 is also much more stable and reliable than Winbind 3.0.33.

To install these packages in RHEL 5.6, you must replace the **samba-common** and **samba-client** packages with **samba3x-common**, **samba3x-client**, and **samba3x-winbind** packages. You will need to remove the all of the samba- packages before you install the samba3x- packages:

```
# yum remove -y samba-common samba-client
# yum install –y samba3x-common samba3x-client samba3x-winbind samba3x-docs
```

It is very important to note that the syntax of the Samba configuration file (/etc/samba/smb.conf) has some important differences between versions 3.0.33, 3.3.8, and 3.5.4. These differences will be covered in the section on Configuring Samba Winbind.

4. Configuring Kerberos

Edit the file `/etc/krb5.conf` to match what is listed below. The critical entries are printed in bold. **VERY IMPORTANT:** In RHEL 5.6 and RHEL 6, the use of DES (des-cbc-crc) encryption is now deprecated in Kerberos 5. You should use ArcFour with HMAC (rc4-hmac) instead. Also note that you must have an entry under the `[domain_realm]` section for your DNS subdomain if the FQDN address of your machine is in a DNS sub-domain below .iastate.edu.

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = IASTATE.EDU
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 7d
forwardable = yes
default_keytab_name = FILE:/etc/krb5.keytab
# default_tkt_enctypes = des-cbc-crc  # DES is deprecated in RHEL 5.6 and 6.
# default_tgs_enctypes = des-cbc-crc  # DES is deprecated in RHEL 5.6 and 6.
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
permitted_enctypes = rc4-hmac

[realms]
IASTATE.EDU = {
kdc = windc1.iastate.edu:88
kdc = windc2.iastate.edu:88
kdc = windc3.iastate.edu:88
kdc = windc4.iastate.edu:88
admin_server = windc1.iastate.edu:749
master_kdc = windc1.iastate.edu
default_domain = iastate.edu
}

[domain_realm]
.iastate.edu = IASTATE.EDU
iastate.edu = IASTATE.EDU
its.iastate.edu = IASTATE.EDU
# It is important to add entries for your DNS subdomain here if
# your host names are in a subdomain of the .iastate.edu DNS domain.
# The example below is for the .engineering.iastate.edu sub-domain.
.engineering.iastate.edu = IASTATE.EDU
engineering.iastate.edu = IASTATE.EDU

[appdefaults]
pam = {
debug = false
ticket_lifetime = 15d
renew_lifetime = 15d
forwardable = true
krb4_convert = false
}
```

Testing the Kerberos Setup

Once you've created /etc/krb5.conf, test to see if Kerberos is set up correctly. Try the following:

```
# kinit andyuser@IASTATE.EDU  [use your account name in place of 'andyuser']
Password for andyuser@IASTATE.EDU:
```

If you entered the password correctly, you should be able to see the details of your Kerberos ticket granting ticket (TGT) using the **klist** command:

```
# klist

Ticket cache:  FILE:/tmp/krb5cc_NNNN_XXX
Default principal:  andyuser@IASTATE.EDU

Valid starting          Expires              Service principal
08/04/10 12:02:51       08/04/10 22:02:55    krbtgt/IASTATE.EDU@IASTATE.EDU
            renew until 08/07/10 12:02:51
```

This shows that you have a valid ticket. In the output above, the ticket is valid for 10 hours and can be renewed up for to 3 days.

5. Configuring Samba Winbind

Much of the essential "glue" for integrating Linux into the Windows domain is provided by Samba. Samba is a set of tools that allow UNIX-based systems to participate in Windows-based services. Winbind is the part of Samba that talks to the Windows Active Directory. Winbind can be used for looking up user and group objects in the Active Directory as well as for checking membership of users in security groups. Note: these instructions only deal with the client-side Samba configurations needed to connect to the Windows environment. We won't cover setting up a Samba server for file and print services here, though many of the configuration settings are identical.

To configure Winbind, we need to set the parameters in the file `/etc/samba/smb.conf`. The parameters will vary somewhat depending on whether you have installed Samba version 3.0.33 or 3.5.4.

Samba 3.0.33 Configuration

Edit the file /etc/samba/smb.conf to match the contents listed below. Critical settings are shown in bold.

```
[global]

    security = ads
    workgroup = IASTATE
    realm = IASTATE.EDU
    wins server = 129.186.142.179, 129.186.142.189
    # The list of domain controllers will be obtained from DNS:
    password server = *

    # Normally, Samba derives the NETBIOS name from the hostname, but you
    # can set it here to be sure:
    netbios name = HYDRA

    encrypt passwords = yes
    use kerberos keytab = yes
    client ntlmv2 auth = yes

    log level = 5
    max log size = 300

 # winbindd configuration
    winbind enum users = no
    winbind enum groups = no
    winbind use default domain = yes
    winbind nested groups = no
    winbind offline logon = false
    winbind nss info = rfc2307

    idmap backend = ad
    idmap uid = 100-1000000
    idmap gid = 100-1000000
```

Samba 3.5.4 Configuration

Edit the file /etc/samba/smb.conf to match what is listed below. Lines beginning with '#' are comments and can be excluded. All other lines are critical.

```
[global]
# The name of the Windows domain
      workgroup = IASTATE

# If not set below, the NetBIOS name is derived from
# the DNS hostname of your machine.
      netbios name =

# Use Active Directory and Kerberos.
      security = ads
      realm = IASTATE.EDU
      encrypt passwords = yes

# Get list of domain controllers from DNS. No need to specify them here.
      password server = *

# The ISU WINS servers:
      wins server = 129.186.142.179, 129.186.142.189

# When client programs such as smbclient connect to Windows services,
# try NTLMv2 authentication.
      client ntlmv2 auth = yes
      encrypt passwords = yes
# Use SASL signing (needed so the domain controllers don't complain).
      client ldap sasl wrapping = sign

# Store host credentials in the kerberos keytab file (/etc/krb5.keytab)
      Kerberos method = system keytab

# Winbind configuration
      idmap config IASTATE: backend = ad
      idmap config IASTATE: range = 100-1000000
      idmap config IASTATE: schema_mode = rfc2307
      idmap backend = tdb
      idmap uid = 1000001 - 1999999
      idmap gid = 1000001 - 1999999

      winbind enum users = No
      winbind enum groups = No
      winbind use default domain = Yes
      winbind nested groups = No
      winbind nss info = rfc2307

 # logging
      log level = winbind:5
      max log size = 300
```

6. Joining the Computer to the Domain

Once the Samba configuration file is set, we can join the Linux system to the Windows domain. This is essentially the same as joining a Windows computer to the domain. There are two parts to the process. First, the computer object must be created in the AD under an OU. Then an OU admin account is used to create the Kerberos host credentials. On the Linux client, we will store the Kerberos host credentials in the key table file `/etc/krb5.keytab`.

If you want, you can pre-create the computer object in the domain using the Active Directory Users and Computer (ADUC) management tool. Or, you can create the computer object and the host credentials in one step using the `net ads join` command from Samba. In either case, you must run the following `net ads join` command on the Linux computer to set the host credentials:

```
# net ads join -U {admin} createcomputer="{OU}" createupn="nfs/hydra.its.iastate.edu@IASTATE.EDU"
```

In the command above, `{admin}` is an OU administrator account with privileges to create computer objects in the OU. The program will prompt you for the password of that account. The `createcomputer="{OU}"` specifies the path to the OU container where the computer object will be stored. For instance, if you want the computer object to be stored under the OU path "ENGR/Computers/Linux", you would enter `createcomputer="ENGR/Computers/Linux"`. You don't have to put your Linux hosts in a separate OU container than your Windows hosts, but it can be easier to manage them that way.

The `createupn=".."` part of the command is used to create the **userPrincipalName** attribute in the Active Directory. As we will explain in the section *Configuring Secure NFS*, the userPrincipalName attribute for a Linux computer object must have a value of the form "nfs/FQDN@IASTATE.EDU" in order for the Secure NFS server to look up the host credentials.

If the net ads join command above is successful (the admin account and password match, and the account has permission to create or modify computer objects under that OU), a copy of the Kerberos host credentials (or keys) will be stored in `/etc/krb5.keytab`. To view the key entries, type:

```
# klist -k
```

You should see results similar to:

```
KVNO  Principal
---- --------------------------------------------------------------
3   host/hydra.its.iastate.edu@IASTATE.EDU
3   host/hydra.its.iastate.edu@IASTATE.EDU
3   host/hydra.its.iastate.edu@IASTATE.EDU
3                   host/HYDRA@IASTATE.EDU
3                   host/HYDRA@IASTATE.EDU
3                   host/HYDRA@IASTATE.EDU
```

```
3                       HYDRA$@IASTATE.EDU
3                       HYDRA$@IASTATE.EDU
3    nfs/hydra.its.iastate.edu@IASTATE.EDU
3    nfs/hydra.its.iastate.edu@IASTATE.EDU
3    nfs/hydra.its.iastate.edu@IASTATE.EDU
```

When any changes to the keytab file are made, you should ensure that permissions on the keytab file are such that only root can read the file:

```
# chmod 400 /etc/krb5.keytab
```

Some Troubleshooting Tips

Always use the 'klist –k' command to check that the host keys have been created properly. In particular, you should have the same key version number in the first column for all the host keys. If not, remove the `/etc/krb5.keytab` file and repeat the 'net ads join ..' command. The key version numbers will increment each time you generate the host principals.

You may occasionally have difficulty getting the host keys generated properly if the computer object was pre-created using the ADUC utility on Windows. In that case, it's easiest to unjoin the computer from the domain, remove the `/etc/krb5.keytab` file and repeat the 'net ads join ..' command. Instead of using the ADUC utility, you can unjoin the computer object using the following Samba command:

```
# net ads leave –U {admin}
```

Where {admin} is the OU administrator account that has permissions to create/delete the computer object.

Adding service to the existing keytab

If additional services need to be added to the existing keytab, the following command can be used:

```
# net ads keytab add <servicename> -U {admin}
```

For example, to add nfs service to hydra.its.iastate.edu issue the command:
net ads keytab add nfs –U {amdin}

Testing the Keytab File

To see if the entries in the keytab file are valid, the root user can use the 'kinit' command to obtain a Kerberos ticket using one the of Kerberos principals listed in `/etc/krb5.keytab`:

```
# kinit -k nfs/hydra.its.iastate.edu@IASTATE.EDU
# klist
Ticket cache: FILE: /tmp/krb5cc_0
Default principal: nfs/hydra.iastate.edu@IASTATE
```

```
   Valid starting       Expires            Service principal
   08/21/10 09:30:58    08/21/10 19:30:58 krbtgt/IASTATE.EDU@IASTATE.EDU
```

Perform another test on the keytab by using the following command to obtain a service principal:

```
# kinit –S host/hydra.its.iastate.edu -k -t /etc/krb5.keytab HYDRA$
```

If this command returns without errors, issue the klist command to view the ticket information:

```
   # klist

Ticket cache: FILE:/tmp/krb5cc_0
Default principal: HYDRA$@IASTATE.EDU

Valid starting     Expires            Service principal
12/04/09 16:23:10  12/05/09 02:23:10 host/hydra.its.iastate.edu@IASTATE.EDU
```

A Cron Job to Re-initialize Host Credentials

Host credentials stored in /etc/krb5.keytab are commonly used by the root user to obtain Kerberos tickets for such purposes as querying LDAP or mounting NFS shares that require Kerberos authentication. Since Kerberos tickets generally have a short lifetime (10 hours by default), it is useful to be able to automatically re-initialize Kerberos tickets for the root user. We can add a couple of **cron** jobs for root that will automatically renew the host credentials on a fixed schedule. The 'crontab –e' command will open the 'vi' editor with the contents of /var/spool/cron/root. Any lines you add will be added to the **cron** table for root when you exit the editor. You may need to ensure that **crond** is running and that it is set to run after a reboot.

```
# crontab -e

[ then enter the following lines ]

# Run kinit to re-initialize host principals at 3am, 11am and, 7pm:

0 3,11,19 * * * value=$RANDOM; while [ $value –gt 300 ] ; do
value=$RANDOM;\ done; sleep $value; /usr/kerberos/bin/kinit –k \
nfs/`hostname`@IASTATE.EDU

@reboot /usr/kerberos/bin/kinit –k nfs/`hostname`@IASTATE.EDU
```

The first **cron** entry above will reinitialize the host credentials using the key "nfs/`hostname`@IASTATE.EDU" at 3am, 11am, and 7pm daily (offset slightly over a 5 minute interval).

The second **cron** entry will re-initialize the credentials immediately after a reboot.

The Security Context of the Kerberos Keytab

If step 6 was successful, a keytab file `/etc/krb5.keytab` with 9 entries should be created. Unfortunately the security context of this file does not match the security context SE-Linux assigns to the keytab files. It simply gets assigned "etc_t" as the command ls –lZ will show:

```
# ls -lZ /etc/krb5.keytab

-rw-------  root root root:object_r:etc_t    /etc/krb5.keytab
```

In order to restore the proper context of the keytab file issue the following command:

```
# restorecon /etc/krb5.keytab
```

Now the ls –lZ command should show the context as "krb5_keytab_t":

```
# ls -lZ /etc/krb5.keytab

-rw-------  root root system_u:object_r:krb5_keytab_t  /etc/krb5.keytab
```

SELinux

SELinux interferes with Samba services, particularly with Winbind. It is strongly advised NOT to disable SELinux but rather add a policy allowing Samba and Winbind services to work properly. Here are the steps to achieve the latter:

Download the policy module file

```
# mkdir /root/SELinux/Samba; cd /root/SELinux/Samba
# wget http://tech.its.iastate.edu/win2000/admin/MySambaNet.pp.gz
# semodule –i MySambaNet.pp
```

7. Start Winbind

Once the computer object is created and properly joined, the Winbind daemon (winbindd) can be started. To start it, do:

```
# service start winbind
# chkconfig winbind on
```

Testing Winbind

To see if Winbind is working, see if you can look up users:

```
# wbinfo –i andyuser
```

Also see if you can get a list of groups in the domain:

```
# wbinfo –g
```

If you do not get a list of groups (it can take several seconds for Winbind to query the entire list), then you should check that your machine is properly joined with the command:

```
# net ads testjoin
```

This command should return with "Join is OK" if your computer is properly joined. If not, check to see that your administrative account has the proper privileges, and that the smb.conf file is set up correctly.

8. Set Up Hesiod

Hesiod is very useful. For many situations, it is much easier to look up basic UNIX information about users and groups using Hesiod than with LDAP or Winbind, so it's a good idea configure Hesiod. Create the file `/etc/hesiod.conf` and enter these lines:

```
rhs=.IASTATE.EDU
lhs=.ns
classes=IN
```

Testing Hesiod

To test if Hesiod is working, we can use the hesinfo command to get the traditional UNIX fields for a user or group. In UNIX, user information is called "passwd" info, and group information is called "group" info:

```
#  hesinfo anyuser passwd
andyuser:*:2060:101:Andy J. User,,,,Andy:/home/andyuser:/bin/tcsh
#  hesinfo users group
users:*:2060:101:Andy J. User,,,,Andy:/home/andyuser:/bin/tcsh
```

9. Modify NSS to use Winbind

The Name Service Switch (NSS) facility on Linux allows you to select the source for looking up user (aka passwd) and group information. In the ISU environment, we can set this source to be either the Active Directory (via Winbind) or Moira (via Hesiod). In order for users and groups to be recognized by Linux, users must have an UID number, and groups must have an assigned GID number. All ISU NetIDs have a UID and all groups set in Moira have a GID assigned. Additionally, all NetIDs plus all Moira groups with the "NFS group" property set are synchronized with the Active Directory. This way, the UID and GID values are available in both the Active Directory and Moira. This means that so-called "bang" users and groups in the Active Directory cannot be used on Linux because they don't have assigned UIDs or GIDs.

Does it matter whether we use Winbind or Hesiod as the source for looking up users and groups? In general, either will work fine. Hesiod has some advantages: it doesn't require running a daemon, and Hesiod will often reflect updates much faster than Winbind. But Winbind has several important advantages in NSS when used to look up groups. Winbind can query membership against protected groups. Hesiod cannot. Winbind 3.5.4 also caches lookups of users and groups while Hesiod does not. Also, Winbind does not suffer from the 16 group limit per user that most UNIX environments impose. For these reasons, Winbind is particularly useful for login access controls, as we will see in the section *Configuring PAM*. We will use Winbind for NSS in the following section.

Edit /etc/nsswitch.conf file so that the `passwd` and `group` entries are set like so:

```
passwd:     files winbind
group:      files winbind
```

Changes to the /etc/nsswitch.conf file are immediate. Once you make the changes shown above, you can use the getent command to see if you can look up users or groups. To look up a user, run the command:

```
#  getent passwd andyuser
andyuser:inactive:2060:101:Andy J. User,,,,Andy:/home/andyuser:/bin/tcsh
```

To look up a group, run the command:

```
# getent group engradmin
engradmin:*:8730:{comma separated list of usernames}
```

This shows the GID of the group "engradmin" is 8730. Note that it is not important at this point that `getent` return the members of the group. It is only important that each group name have a corresponding GID and that the system has a way to look it up.

10. Configuring PAM

The Pluggable Authentication Module (PAM) system is what controls authentication and login access on Linux. The next step in integrating Linux with Windows is to modify the default PAM configuration to use Kerberos authentication for users, and to restrict login access to specific groups.

Edit the file `/etc/pam.d/system-auth` so it contains the following lines (pay attention to their placement in the file; for more information on PAM control keywords refer to the pam man pages):

```
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
# Pardon the line wrap: the following two lines should be on one line
auth        requisite     pam_winbind.so use_first_pass
  require_membership_of=student-lab-acl
auth        requisite     pam_succeed_if.so uid >= 100 quiet
auth        sufficient    pam_krb5.so try_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_succeed_if.so uid < 100 quiet
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet
use_uid
session     required      pam_unix.so
```

This example makes use of two PAM modules: `pam_winbind.so`, and `pam_krb5.so`. The line containing `pam_winbind.so` uses Winbind to check whether the user is a member of a group called `student-lab-acl`. If the user trying to log in is a member of that group, then pam_krb5.so is used check the user's password with Kerberos.

When a Linux user logs in, the system usually requires that the user have a home directory. If they don't, Linux will not create user's home directory automatically. To change this behavior, PAM can be configured to create a home directory for the users in the session configuration.

Open the /etc/pam.d/system-auth file, then scroll down toward the bottom and insert a line before the last line in the session section that reads "session optional map_mkhomedir.so skel=/etc/skel umask=0644" (see below). This line configures PAM to create a home directory for a user if one doesn't exist. It will use the directory /etc/skel as a "skeleton" or template, and it will assign the permissions mask 0644 to the new folder.

```
Session optionalpam_mkhomedir.so skel=/etc/skel umask=0644
```

11. Configuring LDAP for Kerberos (GSSAPI) Authentication

Now that we have host credentials for our computer stored in /etc/krb5.keytab, we can use these credentials to query the LDAP Active Directory database. The capability to query LDAP increases the type of information we can look up about users, groups, and computer objects. First, we need to configure OpenLDAP to query the domain controllers, set the base DN for the domain, and to use the GSSAPI (i.e. Kerberos) SASL mechanism:

Edit the file `/etc/openldap/ldap.conf` so it contains the following lines:

```
#See ldap.conf(5) for details
#This file should be world readable but not world writable

BASE dc=iastate, dc=edu
# All on one line:
URI ldap://windc4.iastate.edu ldap://windc3.iastate.edu
ldap://windc2.iastate.edu ldap://windc1.iastate.edu

SASL_MECH GSSAPI
REFERRALS no

TLS_CACERT /etc/pki/tls/certs/ca-bundle.crt
TLS_REQCERT allow
```

Next, make sure you have a valid Kerberos ticket (use 'klist' to verify). Anyone with a valid Kerberos ticket will be able to use this capability, not just the root user. Then, use the `ldapsearch` command to query information from the Active Directory using LDAP:

To see all of the LDAP attributes for the user andyuser:

```
# ldapsearch -LLL `(samaccountname=andyuser)`
```

This output can be a little lengthy. Sometimes you just want to see the value of a specific attribute. The following will list just the "distinguished name" (aka DN) and the homeDirectory attribute for the user "andyuser":

```
# ldapsearch -LLL '(samaccountname=andyuser)' dn homeDirectory
2>/dev/null | egrep "dn:|homeDirectory:"
```

There is a variety of useful information that can be queried with `ldapsearch`. The ldapsearch man page has several good examples.

12. Configure OpenSSH to use Kerberos Authentication

OpenSSH is commonly used to log in remotely to Linux systems. This section explains how to configure OpenSSH to use Kerberos, including forwardable Kerberos tickets to support "single sign-on". Settings for SSH will need to modified on both the SSH server and the client.

On the SSH Server

Edit the file `/etc/ssh/sshd_config` file on the OpenSSH server machine to contain the following lines:

```
KerberosAuthentication yes
KerberosOrLocalPasswd yes
KerberosTicketCleanup yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

The Kerberos options allow users without Kerberos credentials to log in and get a ticket by presenting the appropriate password. They also allow dealing with ssh clients which do not support Kerberos authentication. The GSSAPI options allow users who have a ticket granting ticket to log in by presenting it instead of password or public key exchange.

Note: When sshd is configured to do Kerberos or GSSAPI authentication, it does not use the PAM authentication settings in /etc/pam.d/system-auth that were covered in the section on *Configuring PAM*. So if SSH is configured to allow GSSAPI authentication, AllowUsers or AllowGroups must be used in /etc/ssh/sshd_config to limit the access. For example:

```
AllowGroups mygroup root
AllowUsers fred sally andyuser root
```

On the SSH Client

In order for "single sign-on" to work, the SSH client will need to request GSSAPI authentication. The following lines in /etc/ssh/ssh_config on the client system will enable GSSAPI authentication:

```
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

With GSSAPI authentication in place on both the server and the client, the last requirement is that a user must request forwardable tickets when they get their initial Kerberos tickets:

```
$ kinit -f
```

With forwardable tickets, a user should be able to connect to a properly configured SSH server without the need to enter their password again:

```
$ ssh hydra.its.iastate.edu
```

13. Secure NFS

One challenge with integrating Windows and Linux is how to do file sharing so that both types of systems can access the data equally. This is complicated by the fact that CIFS is not fully supported by the Linux kernel. One solution is to use a network storage environment that can support both CIFS and Secure NFS access to the same files. The Cyfiles and Researchfiles storage systems at ISU supports both CIFS and Secure NFS. This section describes how to configure Secure NFS on Linux.

There are four things we need to configure:

1) Kerberos host credentials, LDAP, and Winbind as described in previous sections.
2) idmapd, a service used by NFSv4 to map users and groups.
3) rpcgssd, a service that passes kerberos credentials to the NFS server.
4) NFS on the Linux client.

A fifth item, the Linux **automounter**, is an optional service that we will delve into.

Kerberos Host Credentials, LDAP, and Winbind

Due to the way a Secure NFS server grants permissions to mount NFS4 shares, each Secure NFS client must have a user principal of the form:

```
nfs/{fully-qualified-domain-name}@IASTATE
```

If you examine the attributes in the Active Directory for this computer object, you should be able to see that the UserPrincipalName attribute is set to this same principal and the machine must have a valid key for this principal stored in `/etc/krb5.keytab`. Follow the instructions for configuring Samba and joining the client computer to the domain as detailed in section 6, *Joining the Computer to the Domain*. Since Secure NFS uses the same access control permissions as CIFS, it is important for the client to know about the same users and groups as the Windows domain. So it is important to have name services, including DNS, LDAP, Winbind, and Hesiod, configured for the ISU environment.

RPC idmapd

NFS4 uses a service called RPC idmapd to ensure that user and group names are mapped to the same names on the server. Edit the file /etc/idmapd.conf so it contains the following lines:

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = IASTATE.EDU

[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
```

```
[Translation]
Method = nsswitch
```

Then start the `rpcidmapd` daemon and configure it to start at reboot:

```
# service rpcidmapd start
# chkconfig rpcidmapd on
```

RPC gssd

In order for Secure NFS to work, we must run a service called `rpcgssd`. This service must be running before we can mount NFS4 file shares with Kerberos authentication. First, we need to configure some NFS settings on the client. Edit the file `/etc/sysconfig/nfs` so the following two parameters are set as shown:

```
SECURE_NFS=yes
RPCGSSDOPTS="vvv"
```

Then start the `rpcgssd` daemon and configure it to start at reboot:

```
# service start rpcgssd
# chkconfig rpcgssd on
```

The `rpcgssd` daemon doesn't give you a lot of details about what it's doing, but if it starts correctly, you should see a message with the following text in /var/log/messages:

Rpc.gssd[123]: Using (machine) credentials cache: 'MEMORY:/tmp/krb5cc_machine_IASTATE.EDU'

Now you should be able to do a secure mount.

```
# kinit -k "nfs/`hostname`@IASTATE.EDU"
# mount -t nfs4 -o sec=krb5,rw,acl hydra.its.iastate.edu:/sechome/andyuser
/mnt/sechome
```

You can test to see if it's working…

```
# su - andyuser
$ cd /mnt/sechome
$ ls
. Permission denied
```

(A user principal required for read/write)

```
$ kinit
Password for andyuser@IASTATE.EDU

$ls
```

```
Dir1 dir2 dir3
```