Linux Academy

Live! Lab

Use Kerberos to Control Access to NFS Shares

Contents

Kerberos can be used to authenticate NFS shares for added security. This allows users to verify their identity and access the mounted filesystem using their pre-existing Kerberos information.

As with all Kerberos labs, Kerberos requires a full-qualified domain name to work. Update the */etc/hosts* file on all three lab servers to contain the FQDN information:

```
10.0.0.100  kdc-server.mylabserver.com
10.0.0.101  nfs-server.mylabserver.com
10.0.0.102  nfs-client.mylabserver.com
```

Set Up the KDC Server

We want to begin with a configuration that supports our goals with this lab: Authentication, and NFS control. Log into the first server, and `su -` into the *root* user.

Install the Kerberos server, workstation, and PAM Kerberos application:

```
[root@kdc-server ~]# yum install -y krb5-server krb5-workstation pam_
krb5
```

Move to the */var/kerberos/krb5kdc/* directory, which contains two configuration files: *kadm5.acl* and *kdc.conf*.

```
[root@kdc-server ~]# cd /var/kerberos/krb5kdc/
[root@kdc-server krb5kdc]# ls
kadm5.acl  kdc.conf
```

Open *kdc.conf* to update the host names to the appropriate domain name. We are using the domain *mylabserver.com*. Remember to update the */etc/hosts* file so the domain is associated with the private IP address of the server.

```
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88
[realms]
 MYLABSERVER.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-
sha1:normal arcfour-hmac:normal camellia256-cts:normal camellia128-
cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
 }
```

Open the *kadmn5.acl* file, and change *EXAMPLE.COM* to the appropriate domain:

```
/admin@MYLABSERVER.COM *
```

Navigate to the */etc/* directory, and open the *krb5.conf* file. Replace all instances of *EXAMPLE.COM* with *MYHLABSERVER.COM*. Do the same with any lowercase instances. Uncomment required the lines, and ensure the *kdc* server is set to your FQDN for the first server.

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 default_realm = MYLABSERVER.COM
 default_ccache_name = KEYRING:persistent:%{uid}
[realms]
 MYLABSERVER.COM = {
  kdc = kdc-server.mylabserver.com
  admin_server = kdc-server.mylabserver.com
 }
[domain_realm]
 .mylabserver.com = MYLABSERVER.COM
 mylabserver.com = MYLABSERVER.COM
```

Save and exit.

Now, using `kdb5_util`, we want to create the Kerberos database, inputting a master database password when prompted. This may take a few minutes, as the server generates entropy.

```
[root@kdc-server krb5kdc]# kdb5_util create -s -r MYLABSERVER.COM
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm
'MYLABSERVER.COM',
master key name 'K/M@MYLABSERVER.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

Start the service and ensure that the service persists after reboot:

```
[root@kdc-server krb5kdc]# systemctl enable krb5kdc.service kadmin.
service
Created symlink from /etc/systemd/system/multi-user.target.wants/
krb5kdc.service to /usr/lib/systemd/system/krb5kdc.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/kadmin.
service to /usr/lib/systemd/system/kadmin.service.
```

```
[root@kdc-server krb5kdc]# systemctl start krb5kdc.service kadmin.
service
```

We now need to use the Kerberos administration tool to add principals to the KDC configuration. Start `kadmin`.

```
[root@kdc-server krb5kdc]# kadmin.local
Authenticating as principal root/admin@MYLABSERVER.COM with password.
kadmin.local:
```

Create an administrative principal; this is the root user for the system *not* the database password created earlier.

```
kadmin.local:  addprinc root/admin
WARNING: no policy specified for root/admin@MYLABSERVER.COM; defaulting
to no policy
Enter password for principal "root/admin@MYLABSERVER.COM":
Re-enter password for principal "root/admin@MYLABSERVER.COM":
Principal "root/admin@MYLABSERVER.COM" created.
```

Create a user account to test authentications:

```
kadmin.local:  addprinc krbtest
WARNING: no policy specified for krbtest@MYLABSERVER.COM; defaulting to
no policy
Enter password for principal "krbtest@MYLABSERVER.COM":
Re-enter password for principal "krbtest@MYLABSERVER.COM":
Principal "krbtest@MYLABSERVER.COM" created
```

We now need to add the hostname of the server to the Kerberos database, so when authenticating, it is coming from a server.

```
kadmin.local:  addprinc -randkey host/kdc-server.mylabserver.com
WARNING: no policy specified for host/kdc-server.mylabserver.com@
MYLABSERVER.COM; defaulting to no policy
Principal "host/kdc-server.mylabserver.com@MYLABSERVER.COM" created.
```

Now add a local copy of a *keytab* file:

```
kadmin.local:  ktadd host/kdc-server.mylabserver.com
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc-server.mylabserver.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.
```

Quit `kadmin`:

```
kadmin.local:  quit
```

To test our connection, we need to change our SSH configuration. Open the */etc/ssh/ssh_config* file, and uncomment and change the *GSSAPIAuthentication* and *GSSAPIDelegateCredentials* lines to *yes*.

```
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials yes
```

Reload SSH:

```
[root@kdc-server krb5kdc]# systemctl reload sshd
```

Update the authentication configuration to enable Kerberos:

```
[root@kdc-server krb5kdc]# authconfig --enablekrb5 --update
```

Now, if using a firewall, ensure ports *88 TCP/UDP* and *749 TCP* are open.

We then want to locally test our configuration. Add the *krbtest* user:

```
[root@kdc-server krb5kdc]# useradd krbtest
```

`su` into the *krbtest* user:

```
[root@kdc-server krb5kdc]# su - krbtest
```

Initialize the Kerberos configuration:

```
[krbtest@kdc-server ~]$ kinit
Password for krbtest@MYLABSERVER.COM:
```

Run `klist` to view the default principal and ensure it recognizes the principal user:

```
[krbtest@kdc-server ~]$ klist
Ticket cache: KEYRING:persistent:1002:1002
Default principal: krbtest@MYLABSERVER.COM
Valid starting    Expires   Service principal
09/15/2016 12:06:36  09/16/2016 12:06:36  krbtgt/MYLABSERVER.COM@
MYLABSERVER.COM
```

Further confirm by SSHing into the *kdc-server.mylabserver.com* server:

```
[krbtest@kdc-server ~]$ ssh kdc-server.mylabserver.com
```

Configure the Kerberos Client

Because we want to use the third server as the Kerberos client, but the credentials are not included on the Live! Lab page, we need to SSH into the server through one of the other provided servers. Log into the first or second server, then SSH using the private IP:

```
[linuxacademy@kdc-server ~]$ ssh linuxacademy@10.0.0.102
```

Download the Kerberos workstation and the PAM Kerberos application:

```
[root@nfs-client ~]# yum install -y krb5-workstation pam_krb5
```

Now, as before, we need to edit the */etc/krb5.conf*. This should be identical to the */etc/krb5.conf* file used on the Kerberos server.

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 default_realm = MYLABSERVER.COM
 default_ccache_name = KEYRING:persistent:%{uid}
[realms]
 MYLABSERVER.COM = {
```

```
 kdc = kdc-server.mylabserver.com
 admin_server = kdc-server.mylabserver.com
 }
[domain_realm]
 .mylabserver.com = MYLABSERVER.COM
 mylabserver.com = MYLABSERVER.COM
```

Create the *krbtest* user:

```
[root@nfs-client ~]# useradd krbtest
```

Start the Kerberos admin:

```
[root@nfs-client ~]# kadmin
Authenticating as principal root/admin@MYLABSERVER.COM with password.
Password for root/admin@MYLABSERVER.COM:
kadmin:
```

The password is the *root* password created in the prior section.

Add the host:

```
kadmin:  addprinc -randkey host/nfs-client.mylabserver.com
WARNING: no policy specified for host/nfs-client.mylabserver.com@
MYLABSERVER.COM; defaulting to no policy
Principal "host/kdc-client.mylabserver.com@MYLABSERVER.COM" created.
```

Add the keytab:

```
kadmin:  ktadd host/nfs-client.mylabserver.com
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-client.mylabserver.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.
```

Exit `kadmin`:

```
kadmin:  exit
```

As before, update the */etc/ssh/ssh_config* to change the *GSSAPIAuthentication* and *GSSAPIDelegateCredentials* lines:

```
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials yes
```

Save and exit.

Reload SSH:

```
[root@nfs-client ~]# systemctl reload sshd
```

Configure the PAM service:

```
[root@nfs-client ~]# authconfig --enablekrb5 --update
```

Log into the *krbtest* user account and initialize Kerberos:

```
[root@nfs-client ~]# su - krbtest
[krbtest@nfs-client ~]$ kinit
Password for krbtest@MYLABSERVER.COM:
```

Test the client by logging into the KDC server:

```
[krbtest@nfs-client ~]$ ssh kdc-server.mylabserver.com
```

Create an NFS Share

Using our second lab server, we want to make a quick NFS share for use in the lab.

Install the *file-server* group:

```
[root@nfs-server ~]# yum groupinstall -y file-server
```

Ensure your firewall has ports open for NFS.

Enable *rpcbind* and *nsf-server*:

```
[root@nfs-server ~]# systemctl enable rpcbind nfs-server
Created symlink from /etc/systemd/system/multi-user.target.wants/nfs-
server.service to /usr/lib/systemd/system/nfs-server.service.
```

Now, create a directory to share:

```
[root@nfs-server ~]# mkdir /krbdata
[root@nfs-server ~]# chmod 0777 /krbdata/
```

We want to run SELinux to ensure it does not cause any issues when sharing a file off of the root directory. Currently, if you view the detailed information about the directory, we can see that SELinux will have issues with the *:object_r:default_t:s0* permissions:

```
[root@nfs-server ~]# ls -Z /
lrwxrwxrwx. root root system_u:object_r:bin_t:s0   bin -> usr/bin
dr-xr-xr-x. root root system_u:object_r:boot_t:s0  boot
drwxr-xr-x. root root system_u:object_r:default_t:s0   data
drwxr-xr-x. root root system_u:object_r:device_t:s0dev
drwxr-xr-x. root root system_u:object_r:etc_t:s0   etc
drwxr-xr-x. root root system_u:object_r:home_root_t:s0 home
drwxrwxrwx. root root unconfined_u:object_r:default_t:s0 krbdata
lrwxrwxrwx. root root system_u:object_r:lib_t:s0   lib -> usr/lib
lrwxrwxrwx. root root system_u:object_r:lib_t:s0   lib64 -> usr/lib64
drwxr-xr-x. root root system_u:object_r:mnt_t:s0   media
drwxr-xr-x. root root system_u:object_r:mnt_t:s0   mnt
drwxr-xr-x. root root system_u:object_r:usr_t:s0   opt
dr-xr-xr-x. root root system_u:object_r:proc_t:s0  proc
dr-xr-x---. root root system_u:object_r:admin_home_t:s0 root
drwxr-xr-x. root root system_u:object_r:var_run_t:s0   run
lrwxrwxrwx. root root system_u:object_r:bin_t:s0   sbin -> usr/sbin
drwxr-xr-x. root root system_u:object_r:var_t:s0   srv
dr-xr-xr-x. root root system_u:object_r:sysfs_t:s0 sys
drwxrwxrwt. root root system_u:object_r:tmp_t:s0   tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0   usr
drwxr-xr-x. root root system_u:object_r:var_t:s0   var
```

SELinux will prevent this system from mounting. To change this, change the file context using `semanage`:

```
[root@nfs-server ~]# semanage fcontext -a -t public_content_rw_t "/
krbdata(/.*)?"
```

To apply the new settings run a `restorecon`:

```
[root@nfs-server ~]# restorecon -R /krbdata/
```

Finally, we need to edit the */etc/exports* file to add the export, using the third server as the destination, and including permissions options:

```
/krbdata nfsclient.mylabserver.com(rw,no_root_squash)
```

Start the services and export the file system:

```
[root@nfs-server ~]# systemctl start rpcbind
[root@nfs-server ~]# systemctl start nfs-server
[root@nfs-server ~]# exportfs -avr
exporting nfs-client.mylabserver.com:/krbdata
```

Return to the **client system** (server three) and install `nfs-utils`:

```
[root@nfs-client ~]# yum install -y nfs-utils
```

We want to be able to mount the new system:

```
[root@nfs-client ~]# cd /mnt/
[root@nfs-client mnt]# mkdir test
[root@nfs-client mnt]# mount -t nfs nfs-server.mylabserver.com:/krbdata
test
```

Run `df -h` to see your newly-mounted NFS share!

```
[root@nfs-client mnt]# df -h
FilesystemSize  Used Avail Use% Mounted on
/dev/xvda2 10G  1.7G  8.4G  17% /
devtmpfs  3.9G 0  3.9G   0% /dev
tmpfs 3.7G 0  3.7G   0% /dev/shm
tmpfs 3.7G   25M  3.7G   1% /run
tmpfs 3.7G 0  3.7G   0% /sys/fs/cgroup
tmpfs 757M 0  757M   0% /run/user/1001
tmpfs 757M 0  757M   0% /run/user/0
nfs-server.mylabserver.com:/krbdata   10G  1.7G  8.3G  17% /mnt/test
```

However, this is only to test the NFS share. Since we want to work this through Kerberos, unmount the share, then remove the test directory:

```
[root@nfs-client mnt]# umount test
[root@nfs-client mnt]# rm -rf test
```

Protect the NFS Share with Kerberos

With setup finally finished, we need to make some configuration changes on both the Kerberos and NFS servers to allow Kerberos to authenticate and protect NFS.

From the NFS server, add NFS as a principal for Kerberos. Open, install and configure the server as in previous steps, then open `kadmin`:

```
[root@nfs-server ~]# kadmin
Authenticating as principal root/admin@MYLABSERVER.COM with password.
Password for root/admin@MYLABSERVER.COM:
kadmin:
```

As with previous setups, add the host as a principal:

```
kadmin:  addprinc -randkey host/nfs-server.mylabserver.com
WARNING: no policy specified for host/nfs-server.mylabserver.com@
MYLABSERVER.COM; defaulting to no policy
Principal "host/nfs-server.mylabserver.com@MYLABSERVER.COM" created.
```

Now generate the keytab and quit:

```
kadmin:  ktadd host/nfs-server.mylabserver.com
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfs-server.mylabserver.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.
kadmin:  quit
```

Update the SSH configuration, located at */etc/ssh/ssh_config*:

```
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

Reload SSH:

```
[root@nfs-server ~]# systemctl reload sshd
```

Update PAM:

```
[root@nfs-server ~]# authconfig --enablekrb5 --update
```

Our NFS server is now set up as a Kerberos client. However, we still need to add a new principal for NFS. Reopen `kadmin`:

```
[root@nfs-server ~]# kadmin
Authenticating as principal root/admin@MYLABSERVER.COM with password.
Password for root/admin@MYLABSERVER.COM:
```

Add the principal for NFS:

```
kadmin:  addprinc -randkey nfs/nfs-server.mylabserver.com
WARNING: no policy specified for nfs/nfs-server.mylabserver.com@
MYLABSERVER.COM; defaulting to no policy
Principal “nfs/nfs-server.mylabserver.com@MYLABSERVER.COM” created.
```

Regenerate the keytab:

```
kadmin:  ktadd nfs/nfs-server.mylabserver.com
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-server.mylabserver.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.
```

Quit `kadmin`:

```
kadmin:  quit
```

Rerun `authconfig`:

```
[root@nfs-server ~]# authconfig --enablekrb5 --update
```

We now need to update our exports to reflect that we are managing NFS with Kerberos. Open the */etc/exports* file:

```
/krbdata nfs-client.mylabserver.com(rw,no_root_squash,sec=krb5)
```

Notice the added `sec=krb5` segment. This host export requires the host to be registered with Kerberos to use the filesystem.

Export the filesystem:

```
[root@nfs-server ~]# exportfs -avr
exporting nfs-client.mylabserver.com:/krbdata
```

Show the mount:

```
[root@nfs-server ~]# showmount -e localhost
Export list for localhost:
/krbdata nfs-client.mylabserver.com
```

This lets us see that our export list has been exported.

Now, to ensure consistent cooperation between NFS and Kerberos, we need to reboot the NFS server:

```
[root@nfs-server ~]# reboot
```

Once rebooted, ensure the NFS server restarted and the */krbmount* export is available:

```
[root@nfs-server ~]# showmount -e localhost
Export list for localhost:
/krbdata client.mylabserver.com
```

Now, switch to the **client server** (server three). We already have NFS set up on this client, but we still need to register the principals of this host as an NFS client. Open `kadmin`, and add a principal to NFS:

```
[root@nfs-nfs-client ~]# kadmin
Authenticating as principal root/admin@MYLABSERVER.COM with password.
Password for root/admin@MYLABSERVER.COM:
kadmin:  addprinc -randkey nfs/nfs-client.mylabserver.com
WARNING: no policy specified for nfs/nfs-client.mylabserver.com@
MYLABSERVER.COM; defaulting to no policy
```

```
Principal “nfs/nfs-client.mylabserver.com@MYLABSERVER.COM” created.
```

Update the keytab, then exit:

```
kadmin:  ktadd nfs/nfs-client.mylabserver.com
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.
keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfs-client.mylabserver.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.
kadmin:  exit
```

We now need to enable the NFS client target. This is a service that calls the NFS secure service to allow Kerberos to authenticate via NFS:

```
[root@nfs-client ~]# systemctl enable nfs-client.target
[root@nfs-client ~]# systemctl start nfs-client.target
```

At this time, we can mount the system using NFS and Kerberos. Create a test directory, then mount the filesystem:

```
[root@nfs-client mnt]# mkdir krbtest
[root@nfs-client mnt]# mount -t nfs4 -o sec=krb5 nfs-server.mylabserver.
com:/krbdata /mnt/krbtest
```

NFS 4 is the only version of NFS supported by Kerberos 5, so the `-t nfs4` ensures the correct version is used.

Check that the filesystem has mounted by running `df -h`:

```
[root@nfs-client mnt]# df -h
FilesystemSize  Used Avail Use% Mounted on
/dev/xvda2 10G  1.7G  8.4G  17% /
```

```
devtmpfs  3.9G 0  3.9G   0% /dev
tmpfs 3.7G 0  3.7G   0% /dev/shm
tmpfs 3.7G   25M  3.7G   1% /run
tmpfs 3.7G 0  3.7G   0% /sys/fs/cgroup
tmpfs 757M 0  757M   0% /run/user/0
tmpfs 757M 0  757M   0% /run/user/1001
nfs-server.mylabserver.com:/krbdata   10G  1.6G  8.5G  16% /mnt/krbtest
```

We can now write to the mounted directory using our *krbtest* user. Switch users, initialize with `kinit`, then navigate to the mounted directory:

```
[root@nfs-client mnt]# su - krbtest
Last login: Thu Sep 22 10:20:13 EDT 2016 on pts/0
[krbtest@nfs-client ~]$ kinit
Password for krbtest@MYLABSERVER.COM:
[krbtest@nfs-client ~]$ cd /mnt/krbtest/
```

We can also write to the filesystem:

```
[krbtest@nfs-client krbtest]$ echo "test" > textfile.krb.txt
```

The process has been successful!