**Use Kerberos to control access to NFS network shares.**

**Prerequisites**

First, you will have to [configure a KDC](https://www.certdepot.net/rhel7-configure-kerberos-kdc/) (**Kerberos Distribution Center**) called here **kbserver.example.com**.

Then, you will need two additional servers: a **NFS** server (here **nfsserver.example.com**) and a **NFS** client (here **nfsclient.example.com**). If you’ve got only two servers/VM, regroup the **KDC** and the **NFS** servers on the same machine.  
Also, to get **Kerberos** running, [NTP synchronization](https://www.certdepot.net/rhel7-set-ntp-service/) and **hostname resolution** must be working.  
It is advisable to [set up a master DNS server](https://www.certdepot.net/rhel7-configure-master-name-server/) but if none is working, add the following lines in the **/etc/hosts** file of each server (replace the specified ip addresses with yours):

**192.168.1.11 kbserver.example.com**

**192.168.1.12 nfsserver.example.com**

**192.168.1.13 nfsclient.example.com**

**Caution**: When adding a new line in the **/etc/hosts** file, you have to write the fully qualified domain name **just after** the ip address. If you use one or several aliases and add them **before** the fully qualified domain name or if you don’t specify the fully qualified domain name, **Kerberos** will **not** work.

**Kerberos NFS Server Configuration**

Before adding the **Kerberos** configuration, [set up the NFS server](https://www.certdepot.net/rhel7-provide-nfs-network-shares-specific-clients/) (use the **nfsserver.example.com** hostname in this tutorial).

Then, you will have to [add the Kerberos client configuration](https://www.certdepot.net/rhel7-configure-system-authenticate-using-kerberos/) (replace **kbclient.example.com** with **nfsserver.example.com** in this tutorial).

Finally, add the specific **NFS** part to the principals:

# **kadmin**

Authenticating as principal root/admin@EXAMPLE.COM with password.

Password for root/admin@EXAMPLE.COM:

kadmin: **addprinc -randkey nfs/nfsserver.example.com**

WARNING: no policy specified for host/kbclient.example.com@EXAMPLE.COM; defaulting to no policy

Principal "host/nfsserver.example.com@EXAMPLE.COM" created.

Create a local copy stored by default in the **/etc/krb5.keytab** file:

kadmin: **ktadd nfs/nfsserver.example.com**

Entry for principal host/nfsserver.example.com with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type camellia256-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type camellia128-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type des-hmac-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsserver.example.com with kvno 2, encryption type des-cbc-md5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: **quit**

Edit the **/etc/exports** file and add the option **sec=krb5** (or the option that you want, see note):

**/home/tools nfsclient.example.com(rw,no\_root\_squash,sec=krb5)**

**/home/guests nfsclient.example.com(rw,no\_root\_squash,sec=krb5)**

Note1: The **sec** option accepts four different values: **sec=sys** (no **Kerberos** use), **sec=krb5** (**Kerberos** user authentication only), **sec=krb5i** (**Kerberos** user authentication and integrity checking), **sec=krb5p** (**Kerberos** user authentication, integrity checking and **NFS** traffic encryption). The higher the level, the more you consume resources.  
Note2: If you want to use **sec=sys** (no **Kerberos** use), you also need to run **setsebool -P nfsd\_anon\_write 1**

Export the new configuration:

# **exportfs -avr**

exporting nfsclient.example.com:/home/guests

exporting nfsclient.example.com:/home/tools

Check your configuration:

# **showmount -e localhost**

Export list for localhost:

/home/guests nfsclient.example.com

/home/tools  nfsclient.example.com

Activate at boot and start the **nfs-secure-server** service (**RHEL 7.0** only):

# **systemctl enable nfs-secure-server && systemctl start nfs-secure-server**

Note: If you want to get more information in the **/var/log/messages** file, edit the **/etc/sysconfig/nfs** file, assign the **“-vvv”** string to the **RPCIDMAPDARGS**/**RPCSVCGSSDARGS** variables and restart the **nfs-idmap**/**nfs-secure-server** daemons.

**Kerberos NFS Client Configuration**

Before adding the **Kerberos** configuration, [set up the NFS client](https://www.certdepot.net/rhel7-mount-unmount-cifs-nfs-network-file-systems/) (use the **nfsclient.example.com** hostname in this tutorial).

Then, you will have to [add the Kerberos client configuration](https://www.certdepot.net/rhel7-configure-system-authenticate-using-kerberos/) (replace **kbclient.example.com** with **nfsclient.example.com** in this tutorial).

Finally, add the specific **NFS** part to the principals:

# **kadmin**

Authenticating as principal root/admin@EXAMPLE.COM with password.

Password for root/admin@EXAMPLE.COM:

kadmin: **addprinc -randkey nfs/nfsclient.example.com**

WARNING: no policy specified for host/kbclient.example.com@EXAMPLE.COM; defaulting to no policy

Principal "host/nfsclient.example.com@EXAMPLE.COM" created.

Create a local copy stored by default in the **/etc/krb5.keytab** file:

kadmin: **ktadd nfs/nfsclient.example.com**

Entry for principal host/nfsclient.example.com with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type camellia256-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type camellia128-cts-cmac added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type des-hmac-sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/nfsclient.example.com with kvno 2, encryption type des-cbc-md5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: **quit**

Activate at boot and start the **nfs-secure** service (**RHEL 7.0** only):

# **systemctl enable nfs-secure &&** **systemctl start nfs-secure**

Activate at boot and start the **nfs-client** target (**RHEL 7.1** and after):

# **systemctl enable nfs-client.target && systemctl start nfs-client.target**

Note1: Since **RHEL 7.1**, the **nfs-secure** service automatically starts if there is a **/etc/krb5.keytab** file.  
Note2: If you want to get more information in the **/var/log/messages** file, edit the **/etc/sysconfig/nfs** file, assign the **“-vvv”** string to the **RPCIDMAPDARGS**/**RPCGSSDARGS** variables and restart the **nfs-idmap**/**nfs-secure** daemons.  
Note3: With the **RHEL 7.3** release, the **Systemd** init system is able to use aliases. For example, the **nfs.service** is a symbolic link/alias to the **nfs-server.service** service file. This enables, for example, using the **systemctl status nfs.service** command instead of **systemctl status nfs-server.service**.  
Previously, running the **systemctl enable** command using an alias instead of the real service name failed with an error.

Mount the remote directory:

# **mount -t nfs4 -o sec=krb5 nfsserver.example.com:/home/tools /mnt**

Note1: If you get the error message **“mount.nfs4: an incorrect mount option was specified”**, check that you started the correct daemons.  
Note2: It is not necessary to specify the **rw** option, it is done by default.  
Note3: You can test what shares are exported by the **NFS** server with the command **showmount -e nfsserver.example.com** but you first need to stop **Firewalld** on the **NFS** server (or open the **111 udp** and **20048 tcp** ports on the **NFS** server).  
note4: If you don’t specify the **sec** option, the security mechanism will be negotiated transparently with the remote server (see details [here](https://serverfault.com/questions/871899/why-is-o-sec-krb5p-unnecessary-in-the-mount-command)).

To permanently set up the mount, paste the following line in the **/etc/fstab** file:

**nfsserver.example.com:/home/tools /mnt nfs4 sec=krb5**

Switch to the **user01** user:

# **su - user01**

Create a **Kerberos** ticket:

$ **kinit**

Password for user01@EXAMPLE.COM:

Create a file called **testFile**:

$ **cd /mnt**

$ **echo "This is a test." >testFile**

Check the result:

$ **ls -l**

total 8

-rw-rw-r--. 1 user01 user01 16 Sep 7 16:42 testFile