

# Network Administration/System Administration (NTU CSIE, Spring 2024)

## Homework #9

B12902110 呂承諺

April 28, 2024

### 1 No more Frieren Stuff

#### (a) Steps

- (1) Set IP address of the server.

```
$ sudo ip addr add 192.168.30.1/24 broadcast + dev ens4
```

- (2) Set IP address of the client.

```
$ sudo ip addr add 192.168.30.2/24 broadcast + dev ens4
```

- (3) On the server, start the `nfs-server` service.

```
$ sudo systemctl enable --now nfs-server.service
```

- (4) On the client, add the following line to `/etc/fstab`.

```
# /etc/fstab
192.168.30.1:/srv/nfs/share /mnt/nfs-share nfs defaults 0 0
```

- (5) On the client, mount the volume.

```
$ sudo mount -a
```

#### Result

```
$ mount | grep nfs
$ ls -alh /mnt/nfs-share
```

```
[nasa@nfs-client nfs-share]$ mount | grep nfs
192.168.30.1:/srv/nfs/share on /mnt/nfs-share type nfs (rw,relatime,vers=3,rsize=524288,wsiz
e=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=192.168.30.1,mountv
ers=3,mountport=20048,mountproto=udp,local_lock=none,addr=192.168.30.1)
[nasa@nfs-client nfs-share]$ ls -alh /mnt/nfs-share
total 16K
drwxr-xr-x 4 root root 4.0K Feb 27 17:32 .
drwxr-xr-x 3 root root 4.0K Feb 27 14:22 ..
drwx----- 2 nasa root 4.0K Feb 27 15:56 nasa
drwx----- 2 sana root 4.0K Feb 27 15:47 sana
[nasa@nfs-client nfs-share]$
```

## (b) Steps

- (1) On the server, add `no_root_squash` to the exports. Reload the configuration and restart the service.

```
#/etc/exports
/srv/nfs      192.168.30.2(rw,sync,fsid=0,no_root_squash)
/srv/nfs/share 192.168.30.2(rw,sync,no_root_squash)
```

```
$ sudo exportfs -arv
$ sudo systemctl restart nfs-server.service
```

- (2) On the client, create a file under `/mnt/nfs-share`.

```
$ echo "# /mnt/nfs-share/file1.txt" |
    sudo tee /mnt/nfs-share/file1.txt
```

## Result

```
$ ls -alh /srv/nfs/share
```

```
[nasa@nfs-server share]$ ls -alh /srv/nfs/share
total 20K
drwxr-xr-x 4 root root 4.0K Apr 28 19:07 .
drwxr-xr-x 3 root root 4.0K Feb 27 13:22 ..
-rw-r--r-- 1 root root   27 Apr 28 19:07 file1.txt
drwx----- 2 nasa root 4.0K Feb 27 15:56 nasa
drwx----- 2 sana root 4.0K Feb 27 15:47 sana
[nasa@nfs-server share]$
```

## (c) Steps

On the server, create `/srv/nfs/share/asan` and modify the permissions.

```
$ sudo mkdir /srv/nfs/share/asan
$ sudo chown asan /srv/nfs/share/asan
$ sudo chmod 700 /srv/nfs/share/asan
```

## Result

```
$ ls -alh /srv/nfs/share
```

```
[nasa@nfs-server ~]$ ls /srv/nfs/share -alh
total 24K
drwxr-xr-x 5 root root 4.0K Apr 28 19:17 .
drwxr-xr-x 3 root root 4.0K Feb 27 13:22 ..
drwx----- 2 asan root 4.0K Apr 28 19:17 asan
-rw-r--r-- 1 root root   27 Apr 28 19:07 file1.txt
drwx----- 2 nasa root 4.0K Feb 27 15:56 nasa
drwx----- 2 sana root 4.0K Feb 27 15:47 sana
[nasa@nfs-server ~]$
```

## References

- [Network configuration - ArchWiki](#)
- [NFS - ArchWiki](#)
- [nfs\(5\) - Linux man page](#)
- [bash - sudo echo "something" » /etc/privilegedFile doesn't work - Stack Overflow](#)

## 2 Naughty Friend, Sana

- (a) In the AUTH\_SYS authentication method, the user is authenticated at the the client. The client is responsible for telling the server the UID and GID. The server trusts and uses the UID and GID that the client provides. On the server, file permissions and ownership are kept just like any local user.

(b) **Steps**

- (1) On both the server and the client, run `getent passwd asan`.

```
[nasa@nfs-server ~]$ getent passwd asan
asan:x:1002:1002::/home/asan:/usr/bin/bash
```

```
[nasa@nfs-client ~]$ getent passwd asan
asan:x:1010:1002::/home/asan:/usr/bin/bash
```

We discover that the UID doesn't match. It's 1002 on the server and 1010 on the client.

- (2) Change asan's UID on the client to 1002.

```
$ sudo usermod -u 1002 asan
```

### Result

```
[asan@nfs-client nfs-share]$ cd asan
[asan@nfs-client asan]$ pwd
/mnt/nfs-share/asan
[asan@nfs-client asan]$ ls -alh
total 8.0K
drwx----- 2 asan root 4.0K Apr 28 19:17 .
drwxr-xr-x 5 root root 4.0K Apr 28 19:17 ..
[asan@nfs-client asan]$
```

### References

- [9.8. Securing NFS Red Hat Enterprise Linux 6 | Red Hat Customer Portal](#)
- [NFS V4 host authentication](#)

### 3 Nasa Finds krb5

- (a) RPCSEC\_GSS uses the Kerberos protocol for authenticating users. It's based on symmetric-key cryptography, authentication servers, tickets and messages to provide mutual authentication over the Internet. Dedicated servers are responsible for authenticating users and issuing authorizations, so the previous trick would not work.

(b) **Steps**

- (1) Install the **chrony** package on both the server and client for time synchronization. Then start the **chronyd** service.

```
$ sudo pacman -S chrony
$ sudo systemctl enable --now chronyd
```

- (2) Install the **krb5** package on both the server and client.

```
$ sudo pacman -S krb5
```

- (3) Edit **/etc/krb5.conf** on both the server and client to the following.

```
#/etc/krb5.conf
[libdefaults]
    default_realm = NASA.CSIE.NTU

[realms]
# use "kdc = ..." if realm admins haven't put SRV records into DNS
    NASA.CSIE.NTU = {
        admin_server = 192.168.30.1
        kdc = 192.168.30.1
    }

[domain_realm]
    nfs-server = NASA.CSIE.NTU
    nfs-client = NASA.CSIE.NTU

[logging]
    kdc = CONSOLE
```

- (4) Run the following commands on the server.

```
$ sudo kdb5_util create -s
$ sudo systemctl enable --now krb5-kdc.service krb5-kadmind.service
$ sudo kadmin.local
kadmin.local: addprinc nasa
kadmin.local: addprinc asan
kadmin.local: addprinc sana
kadmin.local: addprinc -randkey host/nfs-server
kadmin.local: ktadd -k kbclient.keytab host/nfs-server
kadmin.local: addprinc -randkey nfs/nfs-server
kadmin.local: ktadd -k kbclient.keytab nfs/nfs-server
kadmin.local: addprinc -randkey host/nfs-client
kadmin.local: ktadd -k kbclient.keytab host/nfs-server
kadmin.local: quit
$ sudo scp kbclient.keytab nasa@192.168.30.2:~
```

```
$ sudo install -b -o root -g root -m 600 kbcclient.keytab \  
    /etc/krb5.keytab
```

- (5) Run the following commands on the client.

```
$ sudo install -b -o root -g root -m 600 kbcclient.keytab \  
    /etc/krb5.keytab  
$ kinit  
$ klist  
Ticket cache: FILE:/tmp/krb5cc_1000  
Default principal: nasa@NASA.CSIE.NTU  
  
Valid starting      Expires            Service principal  
04/28/24 12:50:03  04/29/24 12:50:03  krbtgt/NASA.CSIE.NTU@NASA.CSIE.NTU
```

- (6) Edit `/etc/exports` on the server and add `sec=krb5`, and then run `exportfs -arv`.

```
# /etc/exports  
/srv/nfs          192.168.30.2(rw,sync,fsid=0,no_root_squash,sec=krb5)  
/srv/nfs/share    192.168.30.2(rw,sync,no_root_squash,sec=krb5)
```

```
$ sudo exportfs -arv
```

### (c) Result

```
[asan@nfs-client nfs-share]$ getent passwd asan  
asan:x:1010:1002::/home/asan:/usr/bin/bash  
[asan@nfs-client nfs-share]$ cd asan  
[asan@nfs-client asan]$ pwd  
/mnt/nfs-share/asan  
[asan@nfs-client asan]$ ls -alh  
total 8.0K  
drwxr-xr-x 2 1002 root 4.0K Apr 28 13:58 .  
drwxr-xr-x 5 root root 4.0K Apr 28 13:58 ..  
[asan@nfs-client asan]$
```

## References

- [Kerberos \(protocol\) - Wikipedia](#)
- [9.5. Common NFS Mount Options Red Hat Enterprise Linux 6 | Red Hat Customer Portal](#)
- [Kerberos - ArchWiki](#)
- [Get a Kerberos service ticket from the command line - Unix & Linux Stack Exchange](#)
- [NFS & Kerberos not working: export does not exist / Networking, Server, and Protection / Arch Linux Forums](#)
- [\[SOLVED\] Can't mount nfs share / Networking, Server, and Protection / Arch Linux Forums](#)
- [Access denied while mounting nfsv4 with kerberos / Networking, Server, and Protection / Arch Linux Forums](#)