Attaching to NFS Server from Windows

Microsoft Windows can attach to NFS Shares with no problem, though you must use the command line to do it. You also need to install the NFS Client Role (Server) or package (Workstation) as it is not normally turned on. **Note**: everything other than the basic read-only mount requires rebooting the server. This **must** be done for it to work (because it is Microsoft, and . . .)

Read-only access

Attaching is fairly simple, if all you need is Read-Only access. Most web search results only show that. The following would mount **\\192.168.1.5\storage\nfs** on the **n:** drive, but with read only access.

```
mount \\192.168.1.5\storage\nfs n:
```

Read/Write Access as anon (anonymous user)

To get read/write access, you will need to go a step further. If you do not care about mapping Windows users to users on your NFS server, you can just go in as the anonymous user (anon). Note that this will kill all restrictions. Anyone who can connect can read/write/delete any file in the target.

First, you will need to set the anonymous user to have the UID and GID (User ID and Group ID) of the owner of the NFS Server. In many Unix cases under NFSv3, this is root, with a UID and GID of 0. The following PowerShell script will set that. It will require a reboot after the registry keys have been added.

anonymous.ps

```
# Powershell, set anonymous user to owner of nfs shares. Change the -Value to the UID/GID of the Unix user who owns the NFS share
New-ItemProperty HKLM:\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default -Name AnonymousUID -Value 0 -PropertyType "DWord"
New-ItemProperty HKLM:\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default -Name AnonymousGID -Value 0  -PropertyType "DWord"
restart-computer --force
```

After reboot, you can mount the share as the anonymous user with full read/write access. However, remember, everyone can access it the same way.

The **nolock** is very important. Did not research too much, but during testing, ended up with corrupted files when I did not mount using that.

```
# mount drive
```

```
mount -o anon nolock \\192.168.1.5\storage\nfs n:
```

User Mapping

If you want to go to the next step, run regedit and find the following keys and delete them.

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\D
efault\AnonymousUID
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\D
efault\AnonymousGID
```

Then, restart the machine

```
restart-computer --force
```

Create two files, passwd and group, in c:\windows\system32\drivers\etc. These files will map the local Windows user to the remote Unix user. Using this, you can create a unix user, give them ownership of the NFS Share, then, mapping a local Windows user to that remote Unix one, get some semblance of control.

passwd file

This file maps a remote Unix user to the a local Windows user. **Note**: I have not done this yet, so not 100% positive how it works.

passwd

```
# Place in c:\windows\system32\drivers\etc\passwd
# domain\unixuser:x:unixUID:UnigGID:description:c:\Users\WinUser
john:x:1000:100:John,,,:c:\users\john
mary:x:1001:100:Mary,,,:c:\users\mary
```

The passwd file maps the Unix username, UID and GID to a Unix path on this machine. Fields are separated by colons (:). So, the first non-comment line is:

1. **john** - username on the Unix machine
2. **x** - placeholder for password, not used (**x** is placeholder on modern Unix machines stating the password is actually in a file called shadow)
3. **1000** - the UID of john on the NFS server
4. **100** - the GID of john (users) on the NFS server
5. **Rod,,,** - a comment, you can leave this empty with no problem (::)
6. **c:\Users\john** - path to the home directory of user john

Group File

Again, have not tested yet, but this is a group file which maps the local user to a group on the NFS server. In the above case, we are mapping john and mary to group 100, commonly the 'users' group on Unix. Here, I'm simply creating a couple of group names (wheel and users), then putting in the GID for them. I'm unclear on the second number in each entry, as in Unix, that is the name of the group members in many cases. Again, will research and modify when done.

Just like the passwords, this is a colon separated file. The second field is **x** indicating that the password is stored in a shadow file format.

group

```
# place in c:\windows\system32\drivers\etc\group
#group:pass:gid:
wheel:x:0:0
users:x:100:100
```

General Notes

I do not go over them below, but everything is mounted with file access of 755 (read/write owner, read only for everyone else) and case insensitive (standard Windows). The mount is not persistent and will not survive a logout/login. Also, the mount is for the current user only.

The default behavior can be modified with some additional flags to the mount command

- -o fileaccess=777 # gives full read/write to anyone who can get to this
- -o fileaccess=700 # owner can do everything, everyone else can not even read it
- -o casesensitive=yes # makes file access case sensitive, standard Unix

You can also create a startup script to auto-mount the volume when you log in. See the article Windows Autostart Folder for some hints on that.

To make the mapping persistent, there are hints at https://marcoschiavon.net/mount-a-share-for-all-users-on-a-windows-server/ which I have not tried.

Finally, the mount command is your friend. **mount** just by itself will show all mounts with all parameters

Thank you

Want to thank Michael Edie for the clearest description of this I found. See the block post at

https://blog.edie.io/2018/06/16/mounting-nfs-shares-in-windows-using-identity-mapping/

Everyone else that I found was just “oh, install NFS Client, then do a mount,” which is a damned lie, unless you want read/only access. Michael Edie goes deeper into the explanation. Visit that site for info that I decided not to replicate here, with a clearer explanation.

Links

- https://blog.edie.io/2018/06/16/mounting-nfs-shares-in-windows-using-identity-mapping/
- https://www.ibm.com/docs/en/zos/2.4.0?topic=clients-configuring-windows-user-name-mapping
- https://www.golinuxcloud.com/unix-linux-nfs-mount-options-example/
- https://marcoschiavon.net/mount-a-share-for-all-users-on-a-windows-server/ (How to mount a share for all users)

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/microsoft_windows/nfs_client

Last update: **2025/03/27 05:45**