Odne Rindheim

Assignment 6

Task 1: LVM

Created 2 partitions, 5GiB each.. /dev/sda3 and /dev/sda4

Preparing the physical volume:
Sudo pvcreate /dev/sda3

Creating a volume group:
Sudo vgcreate myvg /dev/sda3

Creating some logical volumes:
Sudo lvcreate -L 2GiB lv1 myvg
Sudo lvcreate -L 2GiB lv2 myvg

Activate the LVs:
sudo vgchange -ay myvg

Formatting and mounting LV, making them available to the system:
sudo mkfs.ext4 /dev/myvg/lv1
sudo mkfs.ext4 /dev/myvg/lv2
sudo mkdir /mnt/lv1
sudo mkdir /mnt/lv2

Saving files to the LV:
sudo mount /dev/myvg/lv1 /mnt/lv1
sudo mount /dev/myvg/lv2 /mnt/lv2
Sudo touch /mnt/lv1/file.txt
Sudo touch /mnt/lv2/file.txt

Showing VG size before increasing size:

Odne Rindheim

```
[odnerindheim@localhost ~]$ sudo vgdisplay myvg
  --- Volume group ---
  VG Name               myvg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <5.00 GiB
  PE Size               4.00 MiB
  Total PE              1279
  Alloc PE / Size       1024 / 4.00 GiB
  Free  PE / Size       255 / 1020.00 MiB
  VG UUID               vvOxse-vCdu-dziR-7IEk-ul0e-ObMA-oCVHDf
```

Increasing the volume group size:
sudo pvcreate /dev/sda4
sudo vgextend myvg/dev/sda4
Showing the Volume Group Size after extending:

```
[odnerindheim@localhost ~]$ sudo vgextend myvg /dev/sda4
  Volume group "myvg" successfully extended
[odnerindheim@localhost ~]$ sudo vgdisplay myvg
  --- Volume group ---
  VG Name               myvg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               9.99 GiB
  PE Size               4.00 MiB
  Total PE              2558
  Alloc PE / Size       1024 / 4.00 GiB
  Free  PE / Size       1534 / 5.99 GiB
  VG UUID               vvOxse-vCdu-dziR-7IEk-ul0e-ObMA-oCVHDf
```

Odne Rindheim

Task 2: NFS

I did this task using 2 Virtual machines running Almalinux, with brigaded network settings. It would have been the same process with 2 computers on the lab, but this is more convenient for me since I already have these set up.

Server:

Sudo dnf install nfs-utils
Sudo mkdir -p /share/home
Sudo chmod -R 777 /share/home

Echo “/share/home 192.168.0.173(rw,sync,no_root_squash,no_subtree_check)” | sudo tee -a /etc/exports
Sudo exports -arv
Sudo systemctl enable –now nfs-server

Firewall settings:
Sudo firewall-cmd –zone=public –add-service=nfs –permanent
Sudo firewall-cmd –reload

SELinux configuration:

To allow NFS mounted home directories under SELinux, the following policy needs to be enabled:

Sudo setsebool -P use_nfs_home_dirs 1

Client:

Sudo dnf install nfs-utils
Sudo mkdir -p /share/home
Sudo mount -t nfs 192.168.0.214:/share/home /share/home
Sudo nano /etc/fstab (192.168.0.214:/share/home /share/home nfs defaults 0 0)

Testing:

Signing into a user done from ldap:
Created 2 files, dat151testFile.txt and testfile.txt using touch.

Odne Rindheim

```
[odne@localhost ~]$ ls -l /share/home/odne/
total 0
-rw-r--r--. 1 odne e82 0 Apr 23 15:17 dat151testFile.txt
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Desktop
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Documents
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Downloads
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Music
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Pictures
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Public
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Templates
-rw-r--r--. 1 odne e82 0 Apr 23 15:12 testfile.txt
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Videos
[odne@localhost ~]$ s
```

Checking on the server:

```
[server@localhost ~]$ sudo ls -l /share/home/odne/
total 0
-rw-r--r--. 1 odne e82 0 Apr 23 15:17 dat151testFile.txt
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Desktop
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Documents
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Downloads
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Music
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Pictures
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Public
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Templates
-rw-r--r--. 1 odne e82 0 Apr 23 15:12 testfile.txt
drwxr-xr-x. 2 odne e82 6 Apr 23 15:16 Videos
```

Task 3: Custom PAM profile for pam_mount

Started by creating a new custom profile based on sssd
Sudo authselect create-profile custom-sssd-with-mount -b sssd –symlink-nsswitch

Modified the pam_mount, adding the pam_mount.so, adding the 'with-mount' feature to the profile:
Sudo nano /etc/authselect/custom/custom-sssd-with-mount/password-auth
Sudo nano /etc/authselect/custom/custom-sssd-with-mount/system-auth

auth optional pam_mount.so {include if "with-mount"}
session optional pam_mount.so {include if "with-mount"}

Selecting profile and applying changes:
Sudo authselect select custom/custom-sssd with-mount
Sudo authselect apply-changes

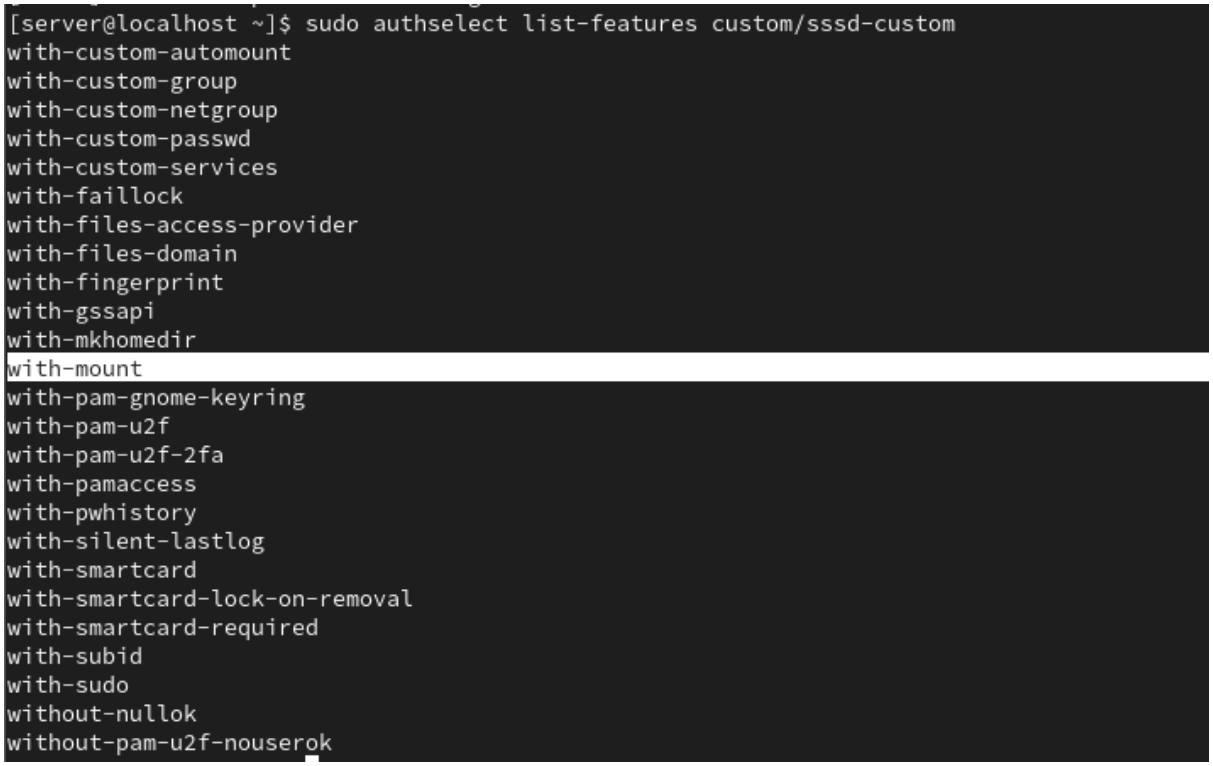
Made changes to the README to include the with-mount and changed the text to say with feature with-mount also.

```
[server@localhost ~]$ sudo authselect enable-feature with-mount
Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.
```

```
[server@localhost ~]$ sudo authselect select custom/sssd-custom with-mount
Profile "custom/sssd-custom" was selected.
The following nsswitch maps are overwritten by the profile:
- passwd
- group
- netgroup
- automount
- services

Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.

[server@localhost ~]$
```

```
[server@localhost ~]$ authselect show custom/sssd-custom
SSSD profile with added feature with-mount
================================================================

with-mount::
   Enable pam_mount to mount volumes for user session
```

Odne Rindheim



Odne Rindheim

Task 4: Decoding PAM rules

The pam_usertype.so with the directive to check if the user is of type “regular”. The control statement preceding the module [default = 1 ignore=ignore success=ok] defines the behavior of the authentication process based on the outcome of this module. If pam_usertype.so confirms the user is regular, the process will continue to the next module in sequence. However, if it does not apply or the check fails, instead of causing an authentication failure, the process will simply skip the next module due to the default=1 action. If the module is irrelevant to the current context, it will be ignored. This setup ensures that the authentication process is flexible and can proceed smoothly even if this check is not critical for all user types or scenarios. It also appears again at line 7.