NFS Server Assignment

WE Innovate X Zero$ploit

Prepared by: Omar Hassan

Supervised by: Eng.Ahmed ElHalwagy

Task:

- *Setting up a NFS server on a machine & connecting to it with another machine.*
- *Using NFS hardening*
- *Log & Detect NFS Access with Auditd*

***Setup** :*

1. Setting Up Kali machine to host NFS server.

Setting	Recommended
RAM	2-4 GB
Disk	10-20 GB
CPU	1-2 Cores
Network	NAT/Bridged

Commands:

Installing NFS server

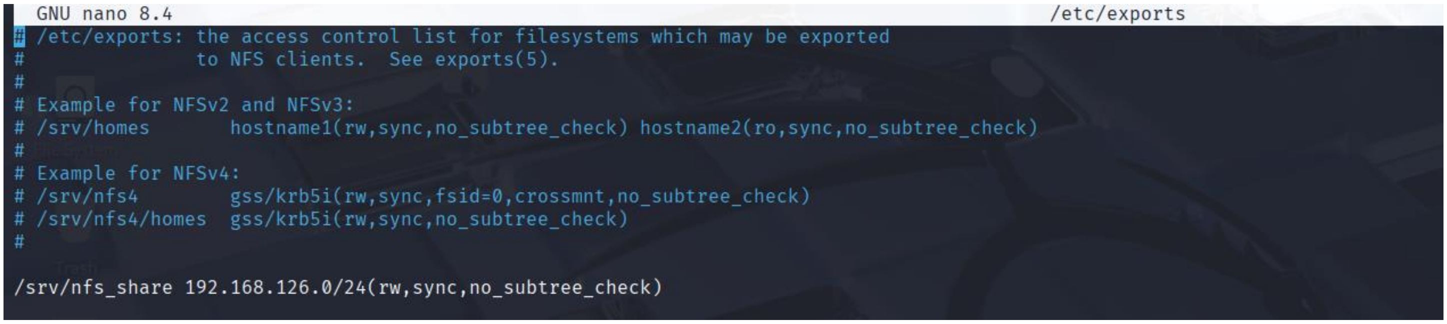
- sudo apt update
- sudo apt install nfs-kernel-server –y

Creating Shared Directory

- sudo mkdir -p /srv/nfs_share\nsudo chown nobody:nogroup /srv/nfs_share\nsudo chmod 755 /srv/nfs_share
- sudo mkdir -p /srv/nfs_share
- sudo chown nobody:nogroup /srv/nfs_share
- sudo chmod 755 /srv/nfs_share

Exporting the share

- sudo nano /etc/exports
- Adding our IP & subnet in the file: /srv/nfs_share 192.168.126.0/24(rw,sync,no_subtree_check)



```
  GNU nano 8.4                                                                                  /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

/srv/nfs_share 192.168.126.0/24(rw,sync,no_subtree_check)
```

Applying Configuration

- sudo exportfs -ra
- sudo exportfs –v

Starting & enabling the service

- sudo systemctl restart nfs-kernel-server
- sudo systemctl enable nfs-kernel-server

Allow through UFW firewall

- sudo apt install ufw
- ip a
- sudo ufw allow from 192.168.126.0/24 to any port nfs
- sudo ufw enable
- sudo ufw status

2. Setting Up CentOS machine to connect to the NFS server.

Setting	Recommended
RAM	1-2 GB
Disk	10-20 GB
CPU	1-2 Cores
Network	NAT/Bridged

Commands:

NFS client on CentOS

- sudo yum install nfs-utils -y

Enabling & starting service

- sudo systemctl enable --now rpcbind
- sudo systemctl start nfs-client.target
- sudo systemctl enable nfs-client.target

Create a mount point

- sudo mkdir -p /mnt/nfs_clientshare

*****Connection** :***

On the NFS Server Machine :

- Retrieve the server IP using command : **ip a**

On the NFS Client Machine :

Commands:

- ping 192.168.126.141
- sudo mount -t nfs 192.168.126.141:/srv/nfs_share /mnt/nfs_clientshare

*****Testing** :***

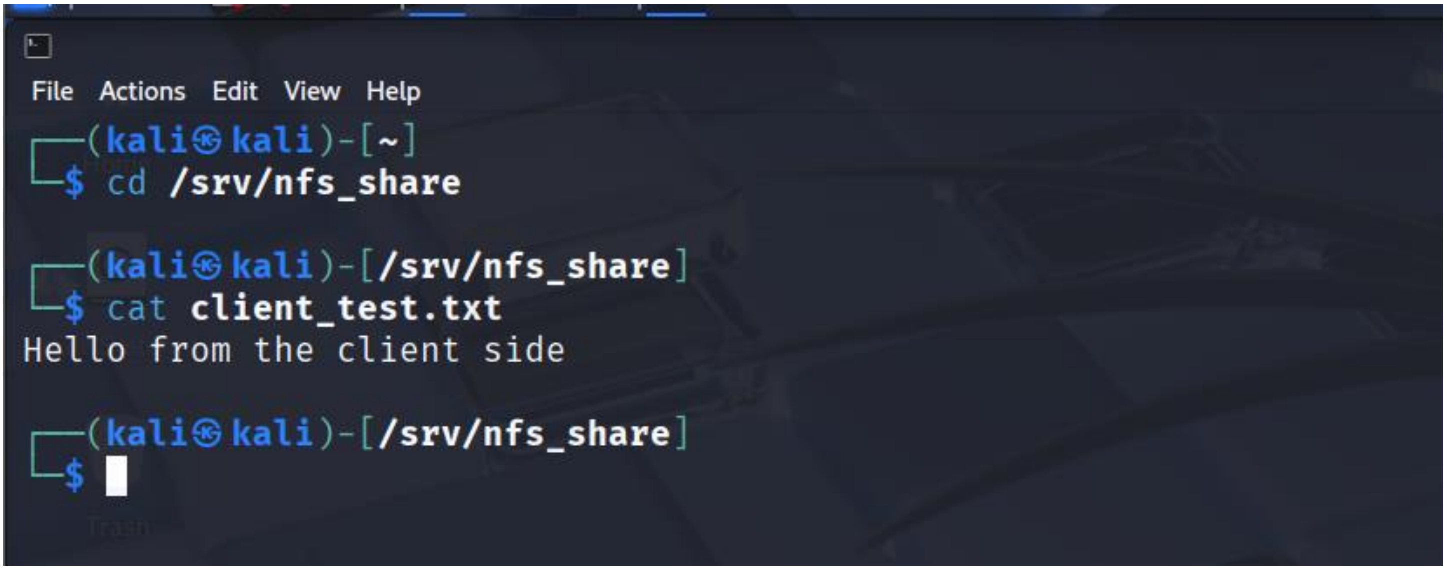
On the NFS Client Machine :

echo "Hello from the client side" | sudo tee /mnt/nfs_clientshare/client_test.txt

On the NFS Server Machine :

- ls /srv/nfs_share
- cd /srv/nfs_share

- cat client_test.txt



```
┌──(kali㉿kali)-[~]
└─$ cd /srv/nfs_share

┌──(kali㉿kali)-[/srv/nfs_share]
└─$ cat client_test.txt
Hello from the client side

┌──(kali㉿kali)-[/srv/nfs_share]
└─$ 
```

*****NFS Hardening** :***

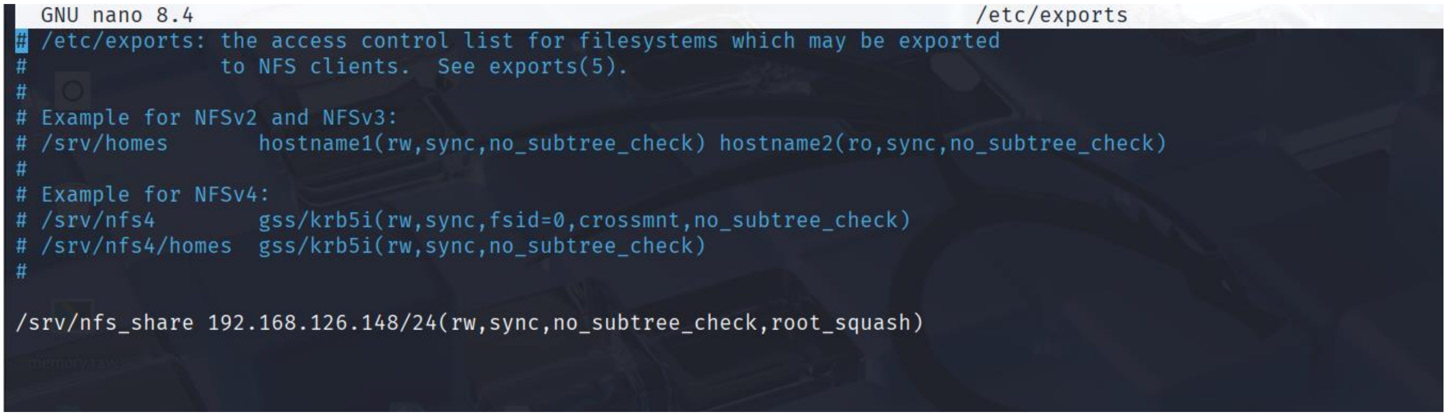
On the NFS Server Machine :

Using root_squash & restricting access tightly to a specific IP

Commands:

- sudo nano /etc/exports
- /srv/nfs_share 192.168.126.148/24(rw,sync,no_subtree_check,root_squash)

This prevents remote users acting as root on the share & only allowing 192.168.126.148 (Client IP) to access this share.



```
  GNU nano 8.4                                                  /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

/srv/nfs_share 192.168.126.148/24(rw,sync,no_subtree_check,root_squash)
```

***Logging** :*

On the NFS Server Machine :

- sudo apt install auditd audispd-plugins -y
- sudo systemctl enable --now auditd
- sudo auditctl -w /srv/nfs_share -p rwxa -k nfs_activity

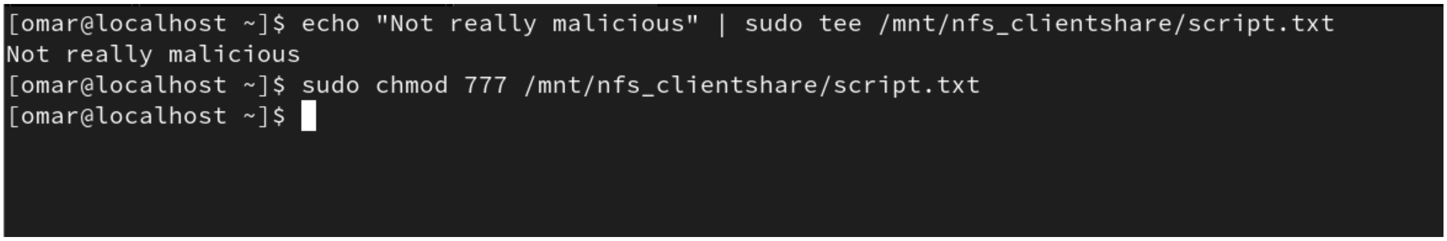
***Simulating a suspicious activity** :*

On the NFS Client Machine :

Scenario : Creating a file and changing it's permissions

Commands:

- echo "Not really malicious" | sudo tee /mnt/nfs_clientshare/script.txt
- chmod 777 /mnt/nfs_clientshare/script.txt

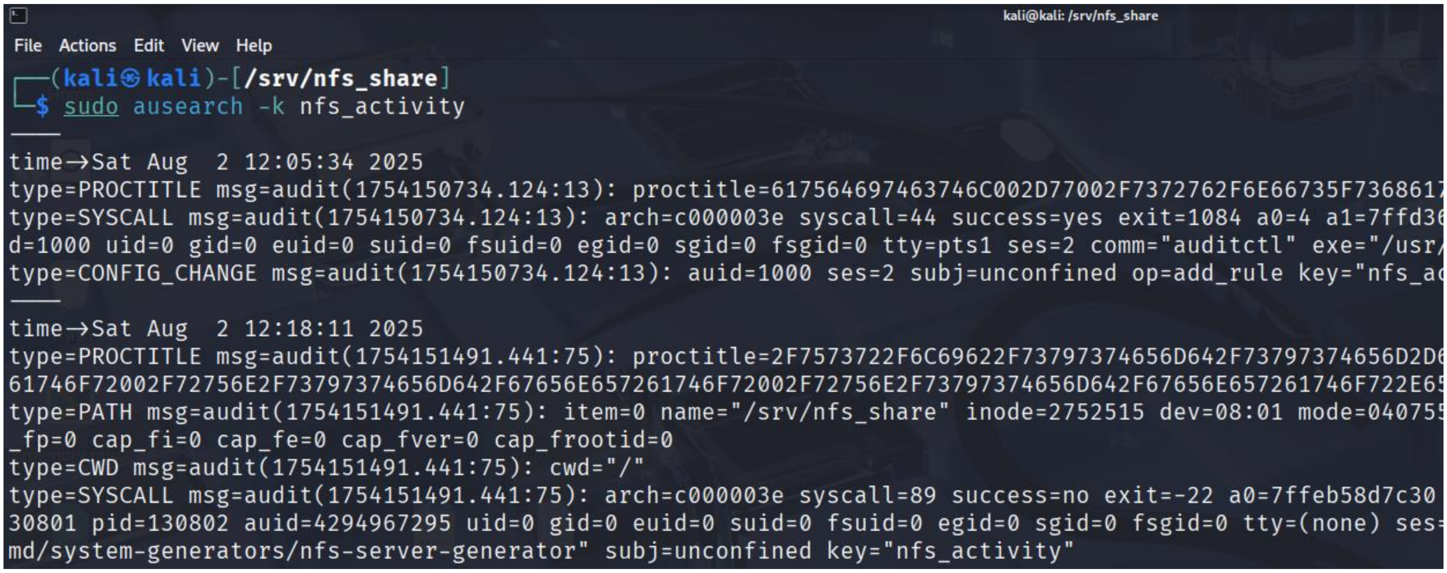


```
[omar@localhost ~]$ echo "Not really malicious" | sudo tee /mnt/nfs_clientshare/script.txt
Not really malicious
[omar@localhost ~]$ sudo chmod 777 /mnt/nfs_clientshare/script.txt
[omar@localhost ~]$ 
```

On the NFS Server Machine :

Checking the logs

- sudo ausearch -k nfs_activity



```
┌──(kali㉿kali)-[/srv/nfs_share]
└─$ sudo ausearch -k nfs_activity
----
time->Sat Aug  2 12:05:34 2025
type=PROCTITLE msg=audit(1754150734.124:13): proctitle=617564697463746C002D77002F7372762F6E66735F7368617
type=SYSCALL msg=audit(1754150734.124:13): arch=c000003e syscall=44 success=yes exit=1084 a0=4 a1=7ffd36
d=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=2 comm="auditctl" exe="/usr/
type=CONFIG_CHANGE msg=audit(1754150734.124:13): auid=1000 ses=2 subj=unconfined op=add_rule key="nfs_ac
----
time->Sat Aug  2 12:18:11 2025
type=PROCTITLE msg=audit(1754151491.441:75): proctitle=2F7573722F6C69622F73797374656D642F73797374656D2D6
61746F72002F72756E2F73797374656D642F67656E657261746F72002F72756E2F73797374656D642F67656E657261746F722E65
type=PATH msg=audit(1754151491.441:75): item=0 name="/srv/nfs_share" inode=2752515 dev=08:01 mode=040755
_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1754151491.441:75): cwd="/"
type=SYSCALL msg=audit(1754151491.441:75): arch=c000003e syscall=89 success=no exit=-22 a0=7ffeb58d7c30
30801 pid=130802 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=
md/system-generators/nfs-server-generator" subj=unconfined key="nfs_activity"
```