

2022-2023



Universidad  
Rey Juan Carlos

## PRÁCTICA SCRIPT

SISTEMAS OPERATIVOS

JOSÉ LUIS MEZQUITA JIMÉNEZ Y MIGUEL ÁNGEL VILLANUEVA

3º INGENIERÍA DE LA CIBERSEGURIDAD  
1 CUATRIMESTRE

## EXPLICACIÓN DEL SCRIPT

En esta práctica de la asignatura de sistemas operativos, hemos tenido que realizar un script. En dicho script hemos tenido que contemplar varios casos ya que, en función de los argumentos que se le pasaba al script, este hacía una u otra cosa. En esta imagen vamos a mostrar donde se encuentra el archivo .sh.

```
jose Luis@jose Luis-VirtualBox:~/practicaScript$ ls
Ejemplo_auth.log  ipLog.sh
jose Luis@jose Luis-VirtualBox:~/practicaScript$
```

El archivo tiene el nombre de ipLog.sh y el fichero Ejemplo\_auth.log, es el que hemos usado de prueba para comprobar en todo momento que estábamos haciendo el script bien.

Para ello, nos hemos creado el fichero con el comando “**touch**” y a continuación, le hemos dado permisos de ejecución al usuario con el siguiente comando “**chmod u+x ipLog.sh**.”

En la siguiente imagen vamos a mostrar el código al completo y después de este analizaremos lo que se hace en todo momento.

```
1 #!/bin/bash
2
3 touch /home/jose Luis/practicaScript/ficheroIP
4
5 ESIP='[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}'
6
7 echo "El script $0 está siendo ejecutado por $USERNAME, al momento de `date`"
8 echo "La versión de bash utilizada es $BASH_VERSION"
9
10
11
12 # con 0 parametros buscar ultimas 100 lineas fichero /var/log/auth.log cual es la ip que mas aparece y el numero de veces
13
14 if test $# -eq 0
15 then
16
17     tail -n 100 /home/jose Luis/practicaScript/Ejemplo_auth.log | grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}' > /
18     home/jose Luis/practicaScript/ficheroIP
19
20     contadorMayor=0
21     contadorAux=0
22     ipAux=0\.\0\.\0\.\0
23
24
25     for IP in $(tail -n 100 /home/jose Luis/practicaScript/Ejemplo_auth.log | grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}')
26     do
27         contadorAux=$(grep -o $IP /home/jose Luis/practicaScript/ficheroIP | wc -l | cut -d ' ' -f 1)
28
29         if test $contadorAux -gt $contadorMayor
30         then
31             contadorMayor=$contadorAux
32             ipAux=$IP
33         fi
34     done
35
36     echo "La ip mas repetida es $ipAux con $contadorMayor veces"
37
38
39
40
41 # con 1 parametro, si es una IP busca esa ip en el /var/log/auth.log y se indica el numero de veces que aparece
42 # con 1 parametro, si es directorio, buscamos los .log contenidos la ultima ip que aparece en el /var/log/auth.log
43
44 elif test $# -eq 1
45 then
46
47     if test -d $1
48     then
49
50         IP=$(grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}' /home/jose Luis/practicaScript/Ejemplo_auth.log |
51         tail -n 1 )
52
53         for FILE in $(find $1 -name "*.log")
54         do
55             grep -H -o $IP $FILE
56         done
57
58     elif [[ $1 =~ $ESIP ]]
59     then
60         grep -E -o $1 /home/jose Luis/practicaScript/Ejemplo_auth.log | wc -l
61
62     else
63         echo "Error: El directorio o la ip no existen"
64         rm /home/jose Luis/practicaScript/ficheroIP
65         exit 3
66     fi
67 fi
```

```

69
70 # con 2 parametros, siendo el 1 el path y el 2 la IP, buscamos la IP dentro de la ruta indicada
71
72 elif test $# -eq 2
73 then
74     if test -d $1 && [[ $2 =~ $ESIP ]]
75     then
76
77         for FILE in $(find $1 -name "*.log")
78         do
79
80             grep -H $2 $FILE
81
82         done
83     else
84         echo "Error: No existen los argumentos u orden incorrecto de argumentos"
85         rm /home/joseluis/practicaScript/ficheroIP
86         exit 2
87     fi
88
89 else
90 # si el numero de argumentos es mayor de 2
91     echo "Error: No has introducido bien los argumentos en $0"
92     rm /home/joseluis/practicaScript/ficheroIP
93     exit 1
94 fi
95
96
97 rm /home/joseluis/practicaScript/ficheroIP
98
99
100

```

Una vez ya hemos mostrado todo el código del script a través de las capturas anteriores, vamos a realizar un análisis más en profundidad del código, en donde se explicará que se ha hecho en todo momento.

En la siguiente captura, se muestra en inicio del script, en donde hemos decidido crear un archivo con el comando **"touch"** en la ruta **/home/joseluis/practicaScript/ficheroIP** en donde guardaremos todos los resultados del script cada vez que se ejecute, es decir, es como crear un fichero auxiliar que nos va a ayudar en todo momento. Un problema que hemos tenido aquí con el que nos hemos peleado bastante hasta rendirnos, es que, a la hora de crear el fichero, nuestra intención era usar un fichero temporal, pero no se nos guardaba ni tampoco se eliminaba después, por lo que optamos por crearnos un fichero normal y luego antes de salir del script, borrarlo con el comando **"rm"**.

En la línea 5 hemos creado un "molde" que lo vamos a usar mas adelante para comprobar si es una ip lo que estamos usando.

En la 7 y 8 estamos imprimiendo información como el usuario, la fecha, la versión del bash y el nombre del script.

```

1 #!/bin/bash
2
3 touch /home/joseluis/practicaScript/ficheroIP
4
5 ESIP='[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}'
6
7 echo "El script $0 está siendo ejecutado por $USERNAME, al momento de `date`"
8 echo "La versión de bash utilizada es $BASH_VERSION"
9
10

```

En la siguiente captura, mostramos lo que hace el script cuando le pasamos 0 parámetros. Cuando no se le pasa parámetros, buscamos en las últimas 100 líneas del fichero /var/log/auth.log cual es la ip que más aparece y el numero de veces. Como no funcionaba con esa ruta, hemos decidido poner como ruta el fichero de prueba que se nos proporcionaba en la práctica.

En la línea 17 cogemos las ultimas 100 líneas con el comando “tail”, luego hacemos un grep donde sacamos solo las IPs, gracias al -o y por último lo incluimos en nuestro ficheroIP creado.

Una vez que tenemos las IPs de las ultimas 100 líneas, nos declaramos variables donde van a estar almacenadas la IP mas repetida hasta el momento y el número de veces.

Después, hacemos un for donde vamos a recorrer todas las IPs que están en las ultimas 100 líneas y vamos a comprobar que no sea mayor, en el momento que la actual es mayor a la acumulada, entramos en el if y lo modificamos.

```
11
12 # con 0 parametros buscar ultimas 100 lineas fichero /var/log/auth.log cual es la ip que mas aparece y el numero de veces
13
14 if test $# -eq 0
15 then
16
17     tail -n 100 /home/joseluis/practicaScript/Ejemplo_auth.log | grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}' > /
home/joseluis/practicaScript/ficheroIP
18
19
20     contadorMayor=0
21     contadorAux=0
22     ipAux=0\.\0\.\0
23
24
25     for IP in $(tail -n 100 /home/joseluis/practicaScript/Ejemplo_auth.log | grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-]
[0-9]{1,3}')
26     do
27         contadorAux=$(grep -o $IP /home/joseluis/practicaScript/ficheroIP | wc -l | cut -d ' ' -f 1)
28
29
30         if test $contadorAux -gt $contadorMayor
31         then
32             contadorMayor=$contadorAux
33             ipAux=$IP
34         fi
35     done
36
37     echo "La ip mas $repetida es $ipAux con $contadorMayor veces"
38
39
40
```

Con 1 argumento, podemos encontrar la posibilidad que sea una IP o un directorio, es por ello, que primero comprobamos si es un directorio (línea 47). En caso de que sea un directorio, buscamos la IP última que aparece en el ejemplo de prueba que se nos proporciona. Después, recorremos todos los archivos .log dentro de esa ruta que se proporciona como argumento, con el -H lo que hacemos es mostrar el nombre del fichero en el que se encuentra esa última ip conseguida anteriormente.

La otra opción es que el argumento pasado sea una IP que va a pasar por nuestro filtro creado al principio del script, en donde si es IP, vamos a buscar el numero de veces que aparece esa IP en el archivo de prueba.

En caso de que no sea ninguna de las anteriores, mostramos el mensaje de error y borramos el fichero auxiliar que nos hemos creado con anterioridad.

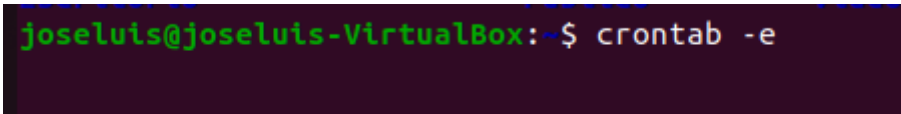
```
40
41 # con 1 parametro, si es una IP busca esa ip en el /var/log/auth.log y se indica el numero de veces que aparece
42 # con 1 parametro, si es directorio, buscamos los .log contenidos la ultima ip que aparece en el /var/log/auth.log
43
44 elif test $# -eq 1
45 then
46
47     if test -d $1
48     then
49
50         IP=$(grep -E -o '[0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}[\./\-][0-9]{1,3}' /home/joseluis/practicaScript/Ejemplo_auth.log |
51         tail -n 1 )
52
53         for FILE in $(find $1 -name "*.log")
54         do
55             grep -H -o $IP $FILE
56         done
57
58     elif [[ $1 =~ $ESIP ]]
59     then
60         grep -E -o $1 /home/joseluis/practicaScript/Ejemplo_auth.log | wc -l
61
62     else
63
64         echo "Error: El directorio o la ip no existen"
65         rm /home/joseluis/practicaScript/ficheroIP
66         exit 3
67     fi
68
```

Con dos parámetros, comprobamos que el primero sea un directorio y que el segundo sea una IP, en caso de que no se cumpla, se imprime mensaje de error, borrando el archivo creado con anterioridad.

En el caso que sea correcto, vamos a recorrer todos los archivos .log en la ruta que se pasa en el argumento \$1 y vamos a mostrar a continuación donde aparece la IP proporcionada en el argumento \$2.

```
68
69
70 # con 2 parametros, siendo el 1 el path y el 2 la IP, buscamos la IP dentro de la ruta indicada
71
72 elif test $# -eq 2
73 then
74     if test -d $1 && [[ $2 =~ $ESIP ]]
75     then
76
77         for FILE in $(find $1 -name "*.log")
78         do
79
80             grep -H $2 $FILE
81
82         done
83     else
84         echo "Error: No existen los argumentos u orden incorrecto de argumentos"
85         rm /home/joseluis/practicaScript/ficheroIP
86         exit 2
87     fi
88
89 else
90 # si el numero de argumentos es mayor de 2
91     echo "Error: No has introducido bien los argumentos en $0"
92     rm /home/joseluis/practicaScript/ficheroIP
93     exit 1
94 fi
95
96
97 rm /home/joseluis/practicaScript/ficheroIP
98
```

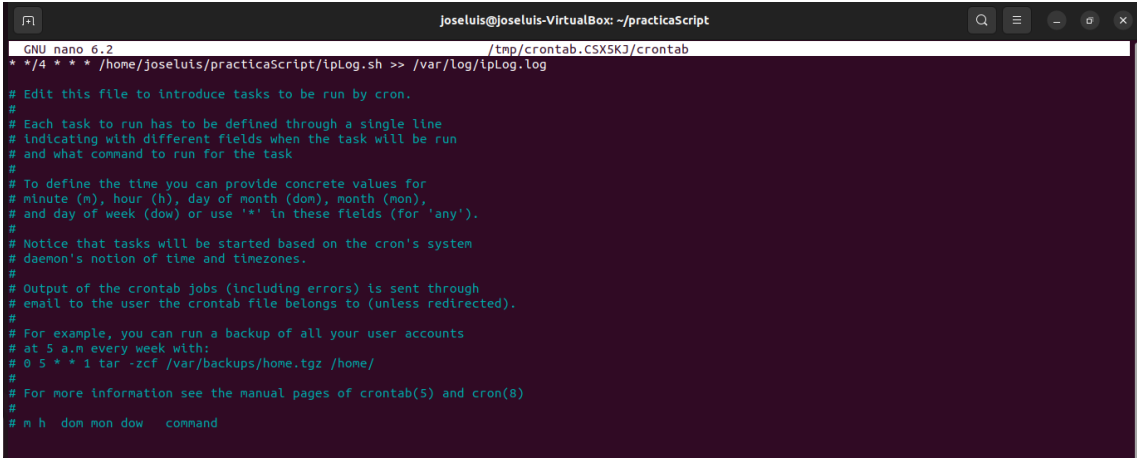
Para que este script se ejecute cada 4 horas, hemos usado el comando “**crontab**”



```
joseluis@joseluis-VirtualBox: ~$ crontab -e
```

En donde se nos abre la siguiente pestaña. En ella, vamos a poner que se ejecute cada 4 horas con el `* */4 * * *` y la ruta del script que queremos que se ejecute.

Por último, ponemos que se guarde el resultado del script en el `/var/log/ipLog.log`

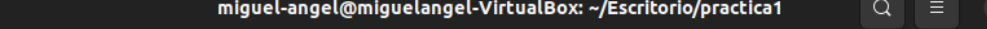


```
joseluis@joseluis-VirtualBox: ~/practicaScript
GNU nano 6.2 /tmp/crontab.CSX5KJ/crontab
* */4 * * * /home/joseluis/practicaScript/ipLog.sh >> /var/log/ipLog.log
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

## PRUEBAS

A continuación, se mostrarán una serie de pantallazos con los resultados que muestra el script en función de los distintos parámetros que le pasemos.

En este caso, no le pasamos ningún argumento:



```
miguel-angel@miguelangel-VirtualBox: ~/Escritorio/practica1
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$ ./ipLog.sh
El script ./ipLog.sh está siendo ejecutado por miguel-angel, al momento de jue 27 oct 2022 11:02:01 CEST
La versión de bash utilizada es 5.1.16(1)-release
La ip mas repetida es 10-77-20-248 con 100 veces
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$
```

Ahora le pasamos un argumento, en este caso, una dirección ip:

```
miguel-angel@miguelangel-VirtualBox: ~/Escritorio/practica1
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$ ./ipLog.sh 10-77-20-248
El script ./ipLog.sh está siendo ejecutado por miguel-angel, al momento de jue 27 oct 2022 11:05:42 CEST
La versión de bash utilizada es 5.1.16(1)-release
7124
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$
```

Al igual que antes le pasamos un argumento, pero esta vez un directorio:

A screenshot of a terminal window titled "miguel-angel@miguelangel-VirtualBox: ~/Escritorio/practica1". The prompt is "miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1\$". The user has entered the command "./ipLog.sh ./". The output shows a message indicating the script is being executed by miguel-angel at 2022-10-27 11:07:01 CEST, followed by the bash version "5.1.16(1)-release". Below this, there are multiple lines of log entries, each starting with ". /Ejemplo\_auth.log:10-77-20-248". The terminal background is dark purple with light green text. There are standard window controls (minimize, maximize, close) in the top right corner.

Ahora le pasamos 2 argumentos, una dirección y una ip:

```
miguel-angel@miguelangel-VirtualBox: ~/Escritorio/practica1
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$ ./ipLog.sh ./ 122.5.240.60
El script ./ipLog.sh está siendo ejecutado por miguel-angel, al momento de vie 28 oct 2022 15:24:04 CEST
La versión de bash utilizada es 5.1.16(1)-release
./Ejemplo_auth.log:Apr 20 13:50:49 ip-10-77-20-248 sshd[3806]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=122.5.240.60 user=root
./Ejemplo_auth.log:Apr 20 13:50:51 ip-10-77-20-248 sshd[3806]: Failed password for root from 122.5.240.60 po
rt 54874 ssh2
./Ejemplo_auth.log:Apr 20 13:51:02 ip-10-77-20-248 sshd[3806]: message repeated 5 times: [ Failed password f
or root from 122.5.240.60 port 54874 ssh2]
./Ejemplo_auth.log:Apr 20 13:51:02 ip-10-77-20-248 sshd[3806]: error: maximum authentication attempts exceed
ed for root from 122.5.240.60 port 54874 ssh2 [preauth]
./Ejemplo_auth.log:Apr 20 13:51:02 ip-10-77-20-248 sshd[3806]: PAM 5 more authentication failures; logname=
uid=0 euid=0 tty=ssh ruser= rhost=122.5.240.60 user=root
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$
```

Si le pasamos más argumentos de lo permitido:

```
miguel-angel@miguelangel-VirtualBox: ~/Escritorio/practica1
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$ ./ipLog.sh ./ 122.5.240.60 arg3
El script ./ipLog.sh está siendo ejecutado por miguel-angel, al momento de vie 28 oct 2022 15:26:13 CEST
La versión de bash utilizada es 5.1.16(1)-release
Error: No has introducido bien los argumentos en ./ipLog.sh
miguel-angel@miguelangel-VirtualBox:~/Escritorio/practica1$
```

## CONCLUSIONES FINALES

Después de la realización completa del script, hemos visto que parte del script hemos podido resolverlo sin mucha dificultad, debido a la realización de los scripts realizados en clase, donde hemos encontrado dificultades ha sido a la hora de ver como obtener las direcciones ip de los archivos log, ya que hemos tenido que emplear el comando 'grep' con una expresión regular extendida diferente a las que hemos empleado en clase, pero gracias a esto hemos conseguido aprender a usar mejor las expresiones regulares en 'grep', además de aprender a usar más opciones de uso del mandato. También hemos aprendido a automatizar scripts, algo totalmente nuevo y que cuenta con muchas opciones de uso. Tras esto, el tiempo empleado en este script ha sido de unas 6 horas, que ha sido el tiempo que esperábamos más o menos.