Práctica1 - Scripting

Sistemas Operativos - Curso 3º
Grado en Ingeniería en Ciberseguridad

Curso 2022-23

1. Descripción

En esta práctica se abordará la utilización del **intérprete de mandatos de Linux**, utilizándolo como herramienta de programación para crear nuevos mandatos sin tener que usar las llamadas al sistema operativo. Es importante hacer notar que este intérprete **nos va a permitir automatiza**r muchas de las **tareas** diarias que son utilizadas **en los sistemas** basados en Linux.

En el grado de ciberseguridad debemos prestar especial atención a los logs del sistema. Los logs en Linux son una fuente básica para saber que está pasando en nuestro equipo. Por lo tanto, todo administrador de sistemas debería saber cómo consultar los logs para poder, de este modo: Detectar las causas de los problemas que puede tener un equipo o servidor, registrar y detectar ataques informáticos, detectar comportamientos anómalos de programas o servicios que tenemos instalados en nuestro equipo, ver el rendimiento de un ordenador o servidor, etc.

/var/log/auth.log: Proporciona un registro de todas las actividades que implican un proceso de autenticación. Por ejemplo, registra los usuarios logueados al sistema operativo. Registra el día, hora, usuario y órdenes que se han ejecutado con el comando sudo, los cronjobs que se han ejecutado, los intentos fallidos de autenticación, etc.

El objetivo de esta práctica es buscar quién está queriendo acceder a mi equipo desde una Ip de manera muy insistente para intentar entrar por ssh o telnet,… o para realizar escaneo de puertos o posibles Denegaciones de Servicio.

En esta primera práctica, **se pide** desarrollar un script denominado **ipLog.sh** que admita uno o varios argumentos. En concreto, la llamada al script tendrá la siguiente forma:

```
ipLog.sh [<path>] [<ip>]
```

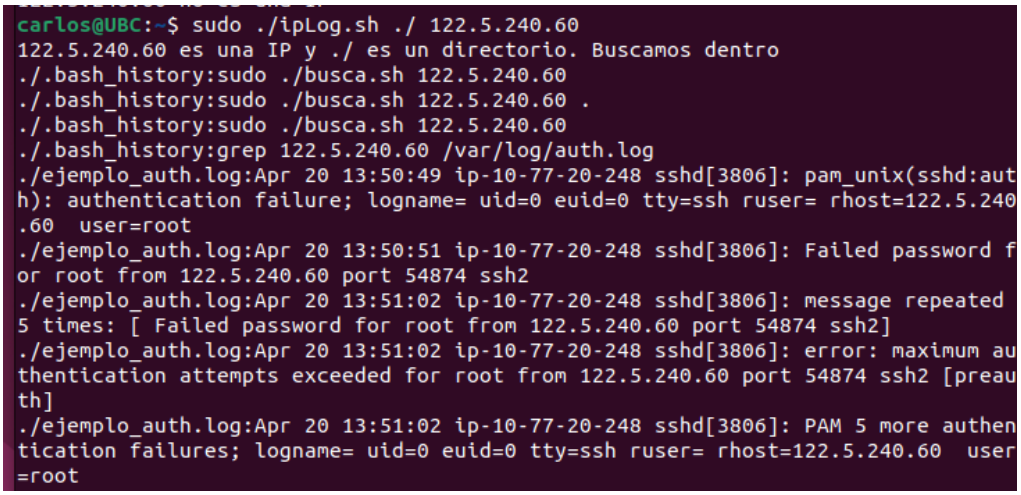
Listado 1: llamada al script

El **objetivo** del script será el siguiente: deberá **buscar** la <ip> indicada como argumento en todos los ficheros de logs del directorio que se indica en el argumento <path> (en caso de que se haya pasado dicho argumento). Además, se han de tener en cuenta los siguientes requisitos:

- Solo se debe buscar en ficheros de logs, evitando el resto de los ficheros
- El script debe saber manejar tanto rutas absolutas como relativas. Es decir, los directorios que se pasen al script pueden ser expresados indicando su ruta absoluta o su ruta relativa.

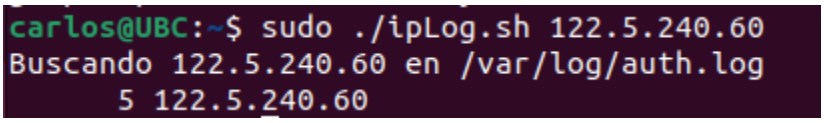
Atendiendo al prototipo de llamada, el script podrá recibir como **argumentos**:

- Dos argumentos: Buscar la IP proporcionada como segundo argumento en los logs contenidos dentro del path indicado. Se mostrarán tantas líneas como IPs aparezcan.



Listado 2: llamada al script . Prueba con 2 argumentos

- Solo argumento IP, en este caso buscaremos la IP en el fichero **/var/log/auth.log** y se indicará el número de veces que aparece.



Listado3: llamada al script . Prueba con argumento IP

- Solo argumento path: Buscaremos en los logs contenidos dentro del path indicado la última IP que aparece en el fichero **/var/log/auth.log.**
  Debemos mostrar el nombre del fichero donde aparece:

```
carlos@UBC:~$ sudo ./ipLog.sh ./
Buscando la IP:0.0.0.0 en el directorio ./
./ejemplo_auth.log:0.0.0.0
./ejemplo_auth.log:0.0.0.0
./ejemplo_auth.log:0.0.0.0
./ejemplo_auth.log:0.0.0.0
./ejemplo_auth.log:0.0.0.0
```

Listado 4: llamada al script . Prueba con argumento path

- Ningún argumento: Buscaremos en las últimas 100 líneas del fichero **/var/log/auth.log** cuál es la IPs que más aparece y el número de veces.

```
carlos@UBC:~$ sudo ./ipLog.sh
Buscando IPs en /var/log/auth.log
      3 0.0.0.0
```

Listado 5: llamada al script . Prueba sin argumentos

El script producirá, por la salida estándar, los mensajes de error y de aviso que se consideren necesarios para la explicación de cada problema.

Nota: tenga en cuenta que se deberá comprobar el valor devuelto por cada mandato utilizado y acabar la ejecución del script notificando el error en el caso de que el mandato haya producido un error irrecuperable.

2. Objetivos parciales

- Comprobación de argumentos. (1 punto)
  - Se debe **comprobar que el número de argumentos recibidos es el correcto**. Si alguno de los directorios pasados como argumentos no existe en el sistema, el programa **no continuará su ejecución** y se mostrará un mensaje de error indicándolo por la salida de error estándar.

- Mensajes de salida. (2 puntos)

- Antes de realizar procesamiento alguno, el script mostrará por la salida estándar: el **nombre** del **usuario** que ejecuta el script, la **fecha y hora** de su ejecución y la **versión de bash** utilizada. Se valorará la gestión de los mensajes de salida en cada una de las funcionalidades de la práctica.

- Gestión de errores. (1 puntos)
  - Se debe **comprobar el valor devuelto** por cada mandato utilizado y acabar la ejecución del script notificando el error, en caso de que el mandato haya acabado con un error .

- Automatizar el script (1 puntos)
  - Automatizar el script para que se pueda ejecutar todos los días cada 4 horas y guardar los resultados en el log: **/var/log/ipLog.log**

- Ejecución correcta de la funcionalidad del script. (5 puntos)

Nota: las puntuaciones para cada objetivo parcial son las puntuaciones máximas que se pueden obtener si se cumplen esos objetivos.

Nota: no se debe hacer un programa separado para cada objetivo, sino un único programa genérico que cumpla con todos los objetivos simultáneamente.

3. Entrega de prácticas

La entrega de prácticas se hará **a través del Campus Virtual** en las fechas anunciadas en el mismo. Se debe entregar **un único archivo resultado de la práctica denominado ipLog.sh**. Dicho fichero debe estar debidamente comentado. Además, se debe entregar una memoria de la práctica en formato pdf. La memoria debe incluir:

- Índice de contenidos
- Autores
- Descripción del script: incluyendo descripción de las principales funciones implementadas.
- Comentarios personales: incluyendo problemas encontrados, críticas constructivas, propuesta de mejoras y evaluación del tiempo dedicado.
- NO DESCUIDE LA CALIDAD DE LA MEMORIA DE SU PRACTICA. Aprobar la memoria es tan imprescindible para aprobar la práctica, como el correcto funcionamiento de esta. Si al evaluar la memoria se considera que no alcanza el mínimo admisible, la práctica se considerará SUSPENSA.

4. Evaluación de la práctica

La práctica se evaluará comprobando el **correcto funcionamiento** de los distintos objetivos, y valorando la **simplicidad del código**, los **comentarios**, la **gestión de errores** y la **calidad de la memoria**. El profesor pedirá una defensa oral de la misma.

5. Autoría de la práctica

La práctica se debe realizar en **grupos de 2 personas como máximo**. El hecho de detectar **copia** en las prácticas expondrá a los alumnos a la **posibilidad de una apertura de expediente disciplinario y expulsión**. Una práctica será considerada copia en caso de que contenga la totalidad o una parte de la práctica de otro alumno. **Se considerará copia en caso de**:

- Archivos que contengan la totalidad o fragmentos de código de otro alumno
- Memorias con la totalidad o fragmentos de frases e imágenes de otro alumno