

Laboratorio de administración y gestión de redes y sistemas

Examen práctico. 13 de enero de 2020.

Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

Preparativos:

- Crea un directorio `~/lagrs.enero.20/`, donde escribirás todo lo indicado a continuación.

Ejercicio 1 (5 puntos)

En este ejercicio escribirás un script en python 3 que, usando la order `netstat`, vigilará ciertos puertos y enviará un mensaje de telegram si algún proceso escucha peticiones en ellos. Edita el fichero `~/lagrs.enero.20/vigila_puertos.py` para que se corresponda con la siguiente especificación:

1. El programa leerá de un fichero de texto con nombre `id_usuario.txt` el identificador de usuario de telegram que recibirá las alarmas. No especifiques ningún *path*, de forma que se busque en el directorio actual.
2. El programa leerá el token del bot de telegram en el fichero `token.txt`. De nuevo, no especifiques ningún trayecto, que el script lea el fichero desde el directorio actual.
3. En las primeras líneas del script define una lista parecida a esta:

```
PUERTOS_TCP = [6666, 443]
```

4. Cada vez que se ejecute el programa, revisará todos los puertos TCP especificados en `PUERTOS_TCP`, y si alguno de ellos está ocupado, en cualquier estado (ya sea *ESCUCHAR*, *ESTABLECIDO*, o cualquier otro en cualquier otro idioma: *LISTEN*, *ESTABLISHED*...), enviará una alarma a través de telegram describiendo toda la información relevante proporcionada por `netstat`.
5. La alarma solo se disparará si el puerto está ocupado en alguna dirección local, es irrelevante lo que pase en direcciones remotas.
6. Usa `romanserver.py` para probar el programa. Cuando funcione, enséñaselo al profesor.

Ejercicio 2 (5 puntos)

En este ejercicio prepararás un contenedor docker que al ejecutarse:

1. El usuario root tendrá una sesión de shell.
2. Este usuario tendrá un directorio `~/bin`.
3. Tendrá en `~/bin/tictac` el siguiente script

```
#!/bin/bash
while true
do
    sleep 1
    echo -n "tic"
    sleep 1
    echo " tac"
done
```

4. La variable de entorno `PATH` del usuario root será la que prepare normalmente la distribución, pero añadiendo el directorio `bin` del `home` del usuario root. De tal forma que se podrá lanzar el script anterior tecleando simplemente `tictac`, desde cualquier directorio, sin añadir ningún `path`.

El contenedor estará construido a partir de `ubuntu 18.04`. No actualices los paquetes y déjalo en inglés. (Si añades paquetes o configuración adicional, tendrás una penalización en la nota). Los ficheros necesarios para esto seguirán el mismo convenio que hemos venido usando durante toda la asignatura.

1. El contenedor estará basado en una imagen llamada `exa`. Si tu nombre de usuario fuera `jperez`, el contenedor se llamaría `jperexa01`. Sustituye `jper` por los primeros 4 caracteres de tu nombre de usuario.
2. Los nombres de los ficheros que debes preparar para esto serán:

```
~/lagrs.enero.20/exa/construye.sh
~/lagrs.enero.20/exa/lanza_jperexa01.sh
~/lagrs.enero.20/exa/context/Dockerfile
~/lagrs.enero.20/exa/context/entrypoint.sh
~/lagrs.enero.20/exa/context/bashrc
~/lagrs.enero.20/exa/context/tictac
```

(Reemplazando `jper` por las primeras 4 letras de tu login)

3. Observa que el fichero de configuración `.bashrc` es un fichero oculto, empieza por punto. Pero cuando lo prepares en el directorio `contexto`, no debe estar oculto (su nombre no empieza por punto). En otras palabras: tienes que preparar un fichero llamado `bashrc`, que dentro del contenedor tendrá por nombre `.bashrc`
4. Enséñaselo al profesor cuando acabes.