

**Laboratorio de Administración y Gestión de Sistemas y Redes**  
**Prueba escrita sobre la teoría. 9 de enero de 2014**  
Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

---

**Instrucciones:**

- Encontrarás en el *home* del puesto del laboratorio el fichero el fichero `~/teoria.tulogin.txt`. Cámbiale el nombre, reemplazando *tulogin* por tu verdadero login. Por ejemplo, si eres *jperez*, debes hacer  

```
mv teoria.tulogin.txt teoria.jperez.txt
```

Revisa que esté bien hecho. Si te equivocas en este paso tan sencillo, suspenderás.
- Dentro del fichero `teoria.jperez.txt`, escribe tu login, nombre y apellidos y contesta al examen.

**Ejercicio 1 (2 puntos)**

¿Qué es una *trap* de SNMP? ¿Qué pasaría si no existiesen las traps ni nada parecido?

**Respuesta**

Es un mensaje asíncrono desde un dispositivo administrado hasta un NMS. Las operaciones *get* y *set* son síncronas, esto es, el NMS hace una solicitud y el dispositivo responde. Mientras que la operación *trap* es asíncrona, la genera el dispositivo cuando sucede *algo importante*, sin que el NMS le *pregunte*. Por tanto, las *traps* resultan similares a las excepciones en los lenguajes de programación

Si no hubiese este tipo de comunicación asíncrona, el NMS no percibiría las situaciones relevantes a menos que preguntase explícitamente por ellas al dispositivo.

**Ejercicio 2 (2 puntos)**

Supongamos que creamos varios usuarios con la orden `adduser`, y posteriormente, por error, borramos el contenido del directorio `/etc/skel`. ¿Qué podría pasar?

**Respuesta**

A los usuarios ya creados esto no les afecta, ya tienen su copia de los ficheros de configuración por omisión. El problema lo tendrían los usuarios creados posteriormente, a quienes les faltarían estos ficheros.

## Ejercicio 3 (2 puntos)

La transparencia 12 del tema *shell II* dice *un intérprete con bit SUID es muy peligroso, normalmente la activación del SUID en un script no tiene efecto*

Explica esto.

### Respuesta

Para el usuario, ejecutar un script o ejecutar un ejecutable parece lo mismo, pero para el sistema es muy distinto. Un ejecutable se ejecuta sin más, pero un script es un fichero de texto plano que no se puede ejecutar por sí mismo. Al intentar ejecutarlo, el sistema operativo comprueba que el fichero no incluye el *número mágico*<sup>1</sup> correspondiente a un ejecutable, pero sí empieza por los *números mágicos* almohadilla admiración, que indican que a continuación el fichero especifica la ubicación de un ejecutable (el intérprete) que es el que realmente se ejecutará (cuando reciba el texto del script)<sup>2</sup>.

Por eso no tiene efecto activar el SUID en un script, porque el script no se ejecuta, se ejecuta el intérprete (el intérprete de shell bash, el intérprete de python o el que corresponda)

Los intérpretes normalmente pertenecen al usuario root, como la mayoría de ficheros *importantes* del sistema. Si un intérprete perteneciente al root tuviera el bit SUID activo, cualquier script que se ejecutase en él tendría permisos de usuario root, con lo que podría hacer cualquier cosa en el sistema. Esto es muy peligroso. Activar el SUID en un fichero ejecutable *normal* puede tener cierto riesgo, pero muy reducido comparado con el caso anterior, porque podemos saber con bastante aproximación qué va a hacer ese programa. Mientras que un intérprete ejecutará cualquier orden que se le pase, es impredecible.

## Ejercicio 4 (4 puntos)

Deseamos que el ejecutable `/usr/local/bin/ups` sea un demonio que monitoriza el sistema de alimentación ininterrumpida del ordenador, y que se ponga en marcha cada vez que se enciende el ordenador. Indica qué ficheros son necesarios y con qué nombre. Describe superficialmente cual será su contenido. Debes seguir todos los convenios del arranque en System V.

### Respuesta

Para seguir correctamente todos los convenios, habrá que renombrar `/usr/local/bin/ups` para que pase a llamarse `/usr/local/bin/upsd`

Luego hará falta un script que maneje el demonio. Se llamaría `/etc/init.d/ups`, aceptaría al menos los parámetros `START`, `STOP` y `RELOAD`. También podría aceptar otros como `RESTART` o `STATUS`. El demonio no tiene por qué manejarse de forma estándar, el script es el que recibe órdenes estándar y envía al demonio órdenes que ya no son estándar, el script *conoce* el interfaz particular de cada demonio concreto.

---

<sup>1</sup>Se llama número mágico a aquel que, por convenio, tiene un significado especial

<sup>2</sup>Si no se indica la ubicación del intérprete, el sistema operativo usará el intérprete por omisión del usuario

En tercer lugar, habrá que hacer que el script se ponga en marcha cuando se encienda el ordenador, y que se detenga al apagarlo. Suponiendo que estemos en una distribución basada en debian, crearemos un fichero dentro del directorio correspondiente al nivel de ejecución por omisión, `/etc/rc2.d`, que será un enlace simbólico apuntando al script descrito en el párrafo anterior y cuyo nombre será `SNNups`, donde NN serán dos dígitos que indican el orden dentro de ese nivel, por ejemplo, `S99ups`.

También será necesario detener el servicio en los niveles 0 y 6 (*Halt* y *Reboot*, respectivamente). Para ello, en los directorios `/etc/rc0.d` y `/etc/rc6.d` crearemos enlaces simbólicos apuntando al script, cuyo nombre será `KNNups`, donde NN representan dos dígitos, por ejemplo `K99ups`.