

# TEMA-3.-Seguridad-de-sistemas.pdf



\_Gxllsi7



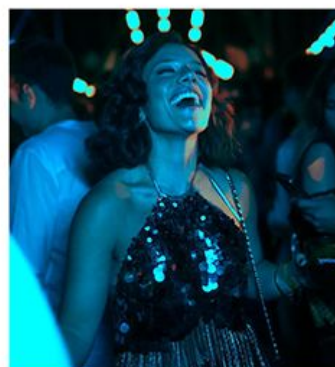
Redes y Seguridad I



3º Grado en Ingeniería Informática



Facultad de Informática  
Universidad Complutense de Madrid



**Todas tenemos una amiga  
experta en recorrer  
kilómetros en discotecas.  
Y si no la tienes eres tú.**

Para ser una experta en kilómetros  
de verdad, déjate guiar por **coches.net**



Compra  
o vende  
tu coche

✓ fácil  
✓ rápido

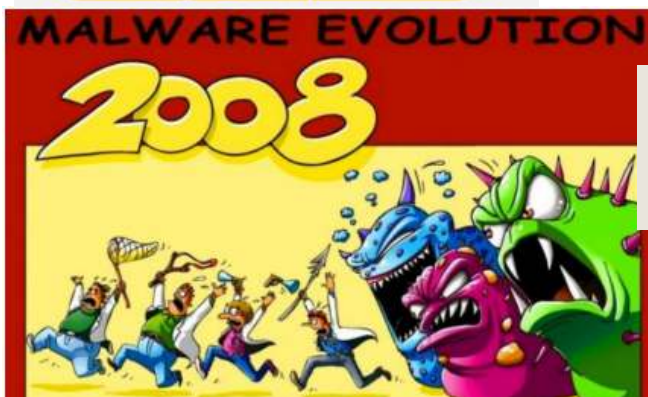
**coches.net**

coches.net



Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.  
**No aprobó nada. Lo probó todo.**

## TEMA 3. Seguridad de sistemas



```
char buf[BUFSIZE];  
gets(buf);
```



Cuando necesites un auténtico experto, déjate guiar por coches.net



WUOLAH

## ÍNDICE

<b>Seguridad del sistema de ficheros.....</b>	<b>3</b>
Sistema de ficheros.....	3
Permisos de acceso.....	3
Cambiar permisos.....	4
Cambiar propietario y grupo.....	4
Listas de control de acceso.....	5
<b>Seguridad de usuarios y grupos.....</b>	<b>6</b>
Sistemas protegidos.....	6
Sistemas multiusuario.....	6
Contraseñas.....	6
Ataques de Contraseña.....	7
Cuentas de usuario.....	8
Contraseñas de usuario.....	9
Grupos.....	9
Cambiar de identidad.....	10
<b>Seguridad de los programas.....</b>	<b>10</b>
Vulnerabilidades software.....	10
Condiciones de carrera.....	11
Memory Overflow.....	11
<b>Malware.....</b>	<b>13</b>
Características.....	13
Tipos.....	14



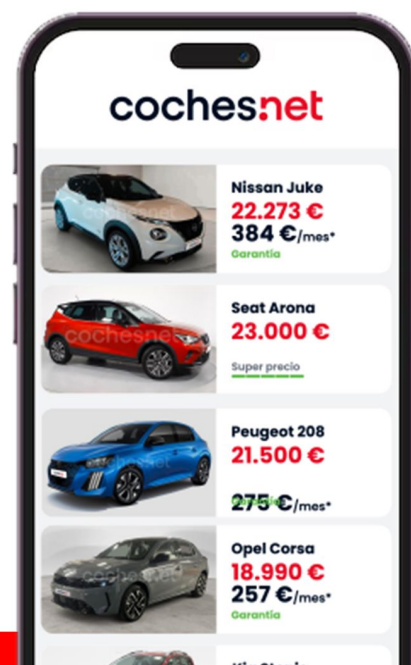
# Nacida para sacar matrículas, experta en tener que matricularme tres veces de lo mismo.

Si quieres ser una verdadera experta, confía en coches.net



Vende o compra  
tu coche

- ✓ Nuevos
- ✓ Renting
- ✓ Km 0
- ✓ Segunda Mano



# Redes y Seguridad I



Banco de apuntes de la

WUOLAH



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



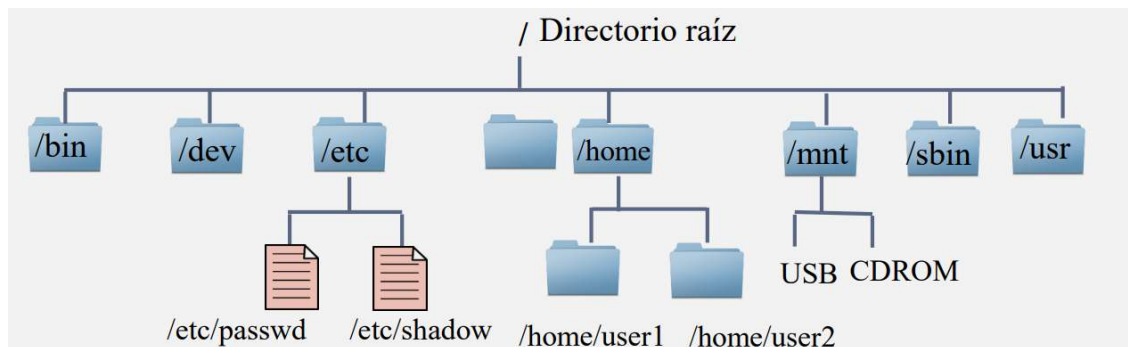
## Seguridad del sistema de ficheros

### Sistema de ficheros

Un sistema de ficheros es la manera que tiene el Sistema Operativo de saber cómo están organizados los ficheros:

- Cómo encontrarlos, ruta (espacio lógico).
- Dónde se encuentran en el espacio físico.
- Qué puede hacer cada usuario con ellos.

Estructura en árbol:



### Permisos de acceso

Los permisos de acceso a ficheros o directorios se definen en términos de acceso a lectura (r), acceso a escritura (w) y acceso a ejecución (x). Para cada fichero o directorio los permisos de acceso se definen para el usuario propietario, el grupo propietario y los demás usuarios del sistema.

```
$ ls -l prueba.txt
-rw-rw-r-- 1 usuario rys 0 2016-03-06 14:52 prueba.txt
```

rw- rw- r--  
usuario grupo otros

Permiso	Ficheros	Directorios
r	Puede ser abierto y leído	Pueden verse sus contenidos (comando ls con todas las opciones) si también x está activo
w	Puede escribirse en él pero no ser renombrado ni borrado	Permite crear, renombrar y borrar ficheros si también x está activo
x	Permite que sea tratado como un programa y se ejecute	Permite el acceso al directorio (con cd)



**Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.**  
**No aprobó nada. Lo probó todo.**

### Cambiar permisos

Para cambiar los permisos de acceso a un fichero o directorio se usa el comando `chmod`. Solo el propietario o el superusuario están autorizados a cambiar los permisos. `chmod` soporta dos modos para especificar los cambios:

- Representación octal.

```
$ chmod 644 texto.txt
$ chmod 100 prueba
```

```
rw- r-- r-- texto.txt
--x --- --- prueba
```

- Representación simbólica:
  - A quién afecta el cambio de permisos: usuario (u), grupo (g), otros (o), todos (a).
  - La operación a ejecutar: añadir (+), eliminar (-), asignar solo ese permiso (=).
  - Qué permisos se van a modificar: r, w, x.

```
rw- r-- r-- script
r-x --- --- prueba
rw- rw- rw- texto.txt
```

```
$ chmod u+x script
$ chmod go=rx prueba
$ chmod o-rw texto.txt
```

```
rw- r-- r-- script
r-x r-x r-x prueba
rw- rw- --- texto.txt
```

Para establecer con qué permisos por defecto se crea un fichero se utiliza el comando `umask`. Usa notación octal para indicar que permisos se eliminan.

```
$ umask 0002
$ touch texto1.txt
$ ls -l texto1.txt
-rw-rw-r-- 1 me me 0 2017-09-06 14:58 texto1.txt
```

### Cambiar propietario y grupo

El comando `chown` permite cambiar el usuario propietario y el grupo de un fichero o directorio:

- Necesita permisos de superusuario. Sintaxis: `chown [user][:[group]] file`
- Ejemplos:
  - `chown pedro texto1.txt`, cambia el usuario propietario a pedro.
  - `chown pedro:rys texto1.txt`, cambia el usuario propietario a pedro y el grupo propietario a rys.
  - `chown :admins texto1.txt`, cambia el grupo propietario al grupo admins.
  - `chown bob: texto1.txt`, cambia el usuario propietario a bob y el grupo propietario al grupo primario de bob.

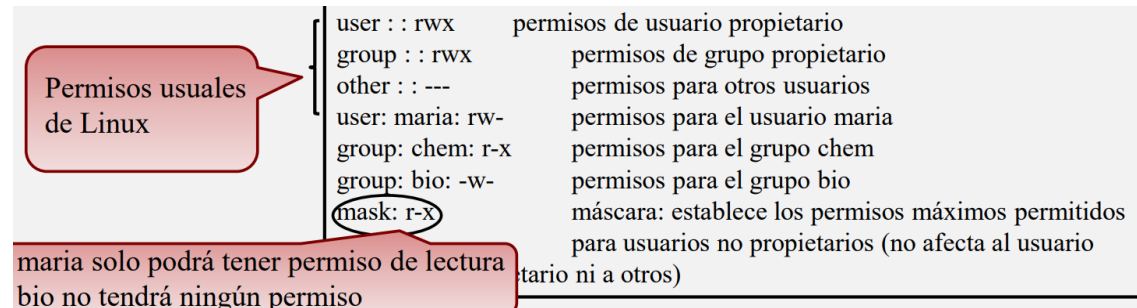
Para cambiar el grupo también se puede usar el comando `chgrp` (`chgrp rys prueba.txt`).





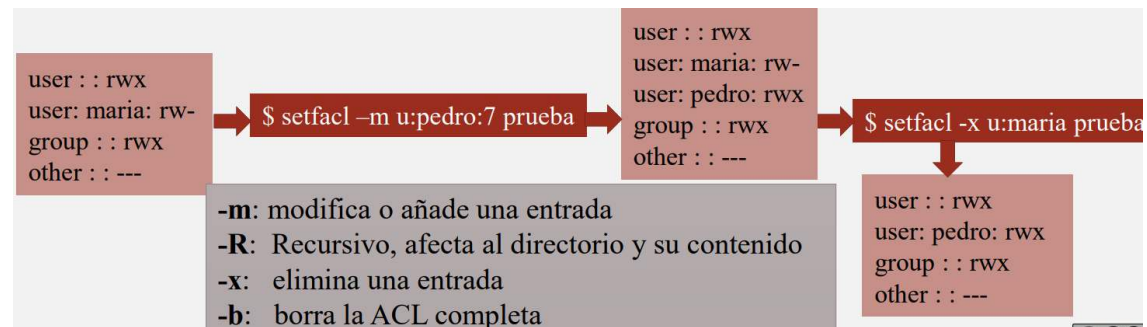
## Listas de control de acceso

Las **ACLs (POSIX Access Control Lists)** permiten definir permisos de acceso a ficheros y directorios para subconjuntos de usuarios o grupos completamente arbitrarios.



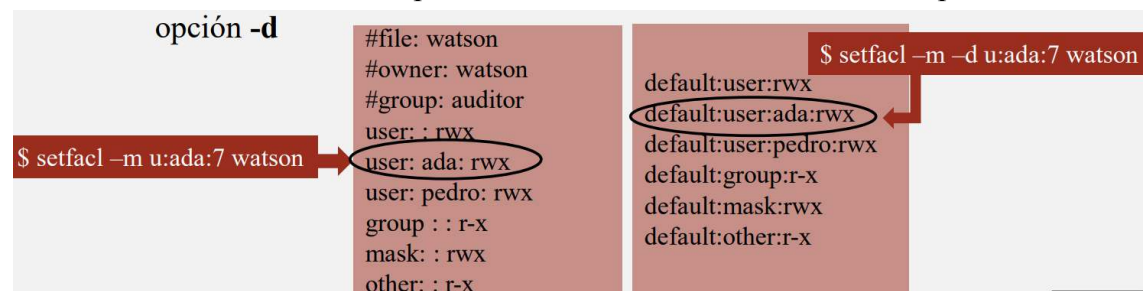
Para trabajar con ACLs se usan los comandos:

- **getfacl:** Nos muestra el contenido de la ACL.
- **setfacl:** Se usa para definir entradas en la ACL.



Se pueden definir ACLs por defecto, solo para directorios:

- Permiten que los ficheros o directorios nuevos que se creen dentro de un directorio hereden los permisos definidos para él.
- Para añadir una entrada por defecto se usa setfacl añadiendo la opción -d.





## Seguridad de usuarios y grupos

### Sistemas protegidos

Es necesario bastionar o proteger un sistema para:

- Reducir al máximo las vulnerabilidades (usuarios, software instalado, servicios de ejecución, SO no actualizado...).
- Reducir el riesgo de fallo humano.
- Minimizar el daño en caso de ataque o fallo.

### Sistemas multiusuario

Las acciones de un usuario no deben:

- Dañar el sistema.
- Modificar o borrar los ficheros de otro usuario.

Será necesario establecer políticas de mínimo privilegio: A cada usuario se le asignarán los permisos mínimos y se le permitirá el acceso solo a las partes del sistema necesarias para realizar su trabajo.

Cuando un usuario abandona una organización se deben cancelar todos sus permisos de acceso.

### Contraseñas

Para acceder al sistema los usuarios tienen que autenticarse, normalmente con una contraseña, entonces si un atacante se hace con la contraseña de un usuario tendrá acceso al sistema. Por ello:

- Las contraseñas deben ser seguras, es decir, difíciles de descubrir.
- Evitar usar datos personales que se pueden obtener mediante ingeniería social: Nombre propio o nombre de alguna persona cercana; Números significativos: cumpleaños, números de teléfono, DNI; Nombres de cosas o personas que nos gustan: película o comida favorita; Nombres de personas o cosas relacionadas con nuestro trabajo.
- Evitar usar palabras que puedan ser reconocidas por programas diseñados para romper contraseñas (p. e. ataques por diccionario): Palabras o partes de una palabra en cualquier idioma; Nombres de lugares, personas famosas, películas, canciones, ...

La robustez de una contraseña la determinan su longitud (la más importante) y su complejidad. Una contraseña segura debería consistir en una cadena lo más aleatoria posible de números, letras y caracteres especiales.



**Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.**  
**No aprobó nada. Lo probó todo.**

Cuando necesites un auténtico experto, déjate guiar por coches.net

### Ataques de Contraseña

- No electrónicos:
  - **Ingeniería social:** Por ejemplo, preguntar al usuario directamente su contraseña haciéndonos pasar por un agente de seguridad.
  - **Shoulder surfing:** Mirar por encima del hombro o desde una esquina qué teclas se pulsan.
  - **Dumpster diving:** Buscar en papeleras contraseñas escritas.
- Activos en línea:
  - **Keylogging:** Proceso que captura las teclas que pulsa el usuario .
  - **Phishing:** La víctima recibe un mail con un enlace malicioso. Al pulsar ese enlace se le pedirá su autenticación en un sistema de confianza y las credenciales introducidas se enviarán al atacante.
- Pasivos en línea: Monitorizar el tráfico para obtener una contraseña en texto claro (telnet).
- Fuera de línea: Se roba el fichero que almacena los hashes de las contraseñas y se intenta descubrir la contraseña.
  - **Fuerza bruta:** Se prueban todos los posibles valores. Si la contraseña es robusta, encontrarla es computacionalmente muy costoso.
  - **Password Spraying.** Fuerza bruta "inteligente":
    - El atacante consigue los nombres de muchas cuentas de usuarios, como cuentas de correo electrónico.
    - Genera un pequeño listado de contraseñas y va probando cada contraseña sobre todas las cuentas obtenidas.
    - Prueba con las contraseñas más utilizadas.
  - **Ataques por diccionario:** Usan listas de palabras. Algunos pueden añadir números y caracteres especiales a las palabras. Otros pueden generar un diccionario a partir de información personal (ingeniería social).
  - **Rainbow Tables:** Usan tablas de hashes computados para las palabras del diccionario. Los hashes de las tablas se comparan con el hash capturado, se ahorra el coste computacional de generar el hash para cada posible contraseña.

**Es necesario proteger las contraseñas.** Los sistemas **NO** guardan las **contraseñas**, sino los **hashes** de las mismas. Si un atacante consigue el fichero donde se guardan los hashes intentará obtener el valor de las contraseñas (ataques de contraseña): Su coste computacional depende del algoritmo de hash utilizado y de la complejidad.



## Cuentas de usuario

Se distinguen tres tipos de usuario:

- **Superusuario (root):**
  - uid 0.
  - Tiene acceso total a todos los archivos y directorios con independencia de propietarios y permisos.
  - Controla la administración de las cuentas de usuario.
  - Puede detener el sistema o cualquier proceso que se esté ejecutando.
  - Instala software en el sistema.
  - Puede modificar o reconfigurar el kernel, controladores, etc.
- **Usuarios especiales:** bin, daemon, adm, lp, sync, ...
  - Son utilizados por programas y servicios del sistema.
  - Por seguridad no tienen todos los privilegios del root.
  - No tienen contraseñas.
  - Se crean automáticamente durante la instalación de Linux o de una aplicación.
  - Generalmente se les asigna un UID entre 1 y 100 (definido en /etc/login.defs)
- **Usuarios normales:**
  - Se usan para usuarios individuales.
  - Cada usuario dispone de un directorio de trabajo, ubicado en /home.
  - Cada usuario puede personalizar su entorno de trabajo.
  - Tienen privilegios completos solo en su directorio de trabajo.
  - Se les asigna normalmente un uid superior a 1000 (definido en /etc/login.defs).

La información sobre las cuentas de usuario se guarda en el fichero /etc/passwd. Este fichero tiene una entrada para cada cuenta de usuario:

username:x:uid:gid:user information:home-directory:login-shell

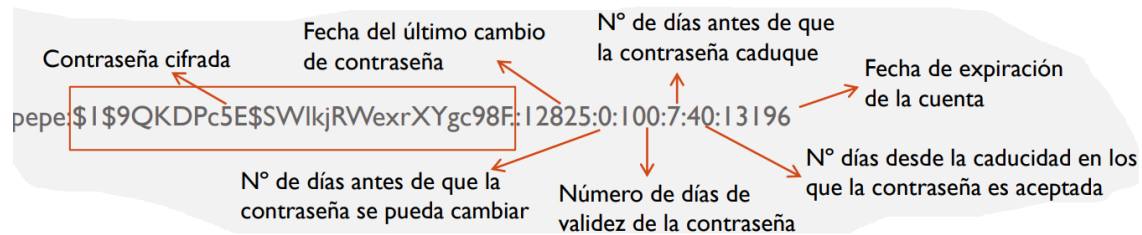
Campo usado para la contraseña, si está a x indica que la contraseña va cifrada en el fichero /etc/shadow

Se modifica cada vez que se crea o se elimina una cuenta de usuario. Los usuarios normales solo tienen permiso de lectura de este fichero. Ejemplo:

maria:x:1000:1000:Maria Lopez, despacho25,,:/home/maria:/bin/bash

## Contraseñas de usuario

Las contraseñas cifradas se almacenan en el fichero /etc/shadow. Solo el superusuario tiene permisos de acceso a este fichero:



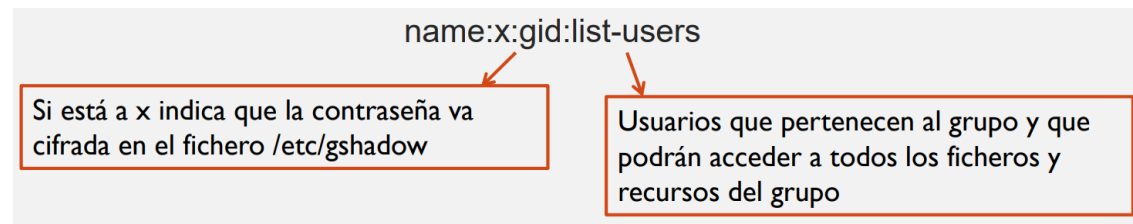
- Formato de la contraseña en el fichero /etc/shadow: \$id\$salt\$hashed.
- \$id\$ indica el algoritmo utilizado para generar el hash:
  - \$1\$: MD5
  - \$2\$: Blowfish
  - \$5\$: SHA-256
  - \$6\$: SHA-512
  - \$y\$: Yescrypt

## Grupos

Conjunto de usuarios que comparten ficheros y otros recursos del sistema. Cada grupo tiene asignado un identificador. Existen dos tipos de grupos:

- **Grupo primario:** Es el grupo que se aplica a un usuario cuando accede a su cuenta (Nombre grupo=nombre de usuario y uid=gid).
- **Grupo secundario:** Cualquier otro grupo adicional al que puede pertenecer un usuario.

Todos los grupos deben aparecer en el fichero /etc/group:



Ejemplo: lab5:x:1000:maria,jorge,pedro,ana





**Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.  
No aprobó nada. Lo probó todo.**

### Cambiar de identidad

Dentro de una sesión de shell existen dos formas de asumir la identidad de otro usuario:

- **El comando su:** Permite asumir la identidad de otro usuario e iniciar una nueva sesión de shell con la identidad de ese usuario.
  - Se asumirán todos los permisos de la nueva identidad.
  - Si usamos la opción - : \$ su - [usuario], en la nueva shell se cargan las variables de entorno y el directorio de trabajo del nuevo usuario.
  - Si se omite el nombre de usuario se asumirá la identidad del superusuario \$ su -.
  - Para que se abra la nueva shell será necesario introducir la contraseña del usuario cuya identidad se quiere asumir.
- **El comando sudo:** Permite a un usuario asumir la identidad de otro, aunque puede tener restringidos algunos privilegios del nuevo usuario. El superusuario configura el fichero /etc/sudoers para especificar qué comandos pueden ejecutar ciertos usuarios cuando asumen una determinada identidad. No se inicia una nueva shell. Será necesario introducir la propia contraseña, no la del nuevo usuario.

### Seguridad de los programas

El SW normalmente se desarrolla **pensando principalmente en la funcionalidad** y no en la seguridad. La **seguridad suele ser una preocupación posterior** y no está presente en el momento del desarrollo del código. Esto hace que podamos terminar creando programas con importantes agujeros de seguridad.

### Vulnerabilidades software

Una vulnerabilidad software se define como un error en la especificación, desarrollo o configuración del software que hace que su explotación pueda violar (de forma explícita o implícita) la política de seguridad de un sistema.

- **Violaciones de seguridad de la memoria:** Buffer overflows, Dangling pointers.
- **Errores de validación de entrada:** Format string attacks, SQL injection, Code injection, E-mail injection, HTTP header injection, ...
- **Condiciones de carrera:** Time-of-check-to-time-of-use bugs, Symlink races.
- **Errores de confusión de privilegios:** Cross-site request, Clickjacking, escalada de privilegios, ...
- **Errores de la interfaz de usuario:** Obligar al usuario a tomar una decisión de seguridad que no le compete, ...

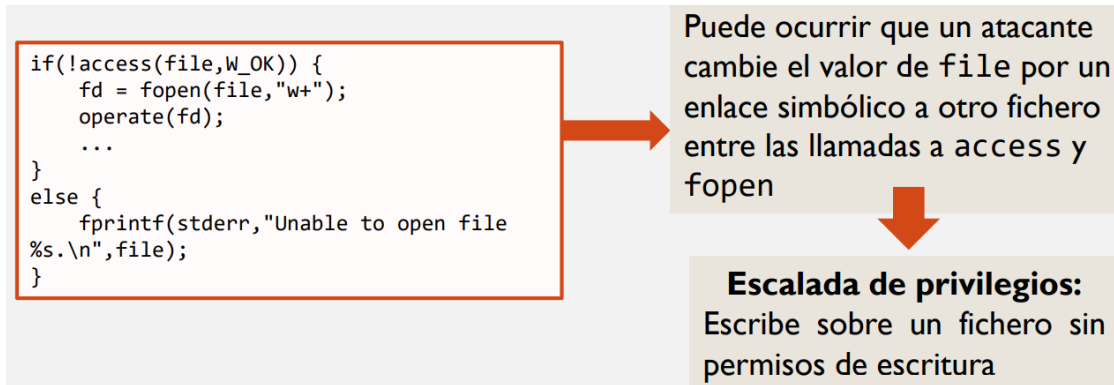
**IMPORTANTE SABER CONCEPTOS DE SISTEMAS OPERATIVOS**



## Condiciones de carrera

Un código, que se ejecuta concurrentemente con otro código, requiere acceso exclusivo a un cierto recurso compartido (p. e. una variable), y debido a la falta de sincronización, este recurso es modificado por el otro código.

- **Time-of-check-to-time-of-use (TOCTOU):** El software comprueba el estado de un recurso antes de usarlo, pero el estado del recurso puede cambiar entre la comprobación y su uso. Ejemplo:



Posibilidad de explotación (CWE): **Media**

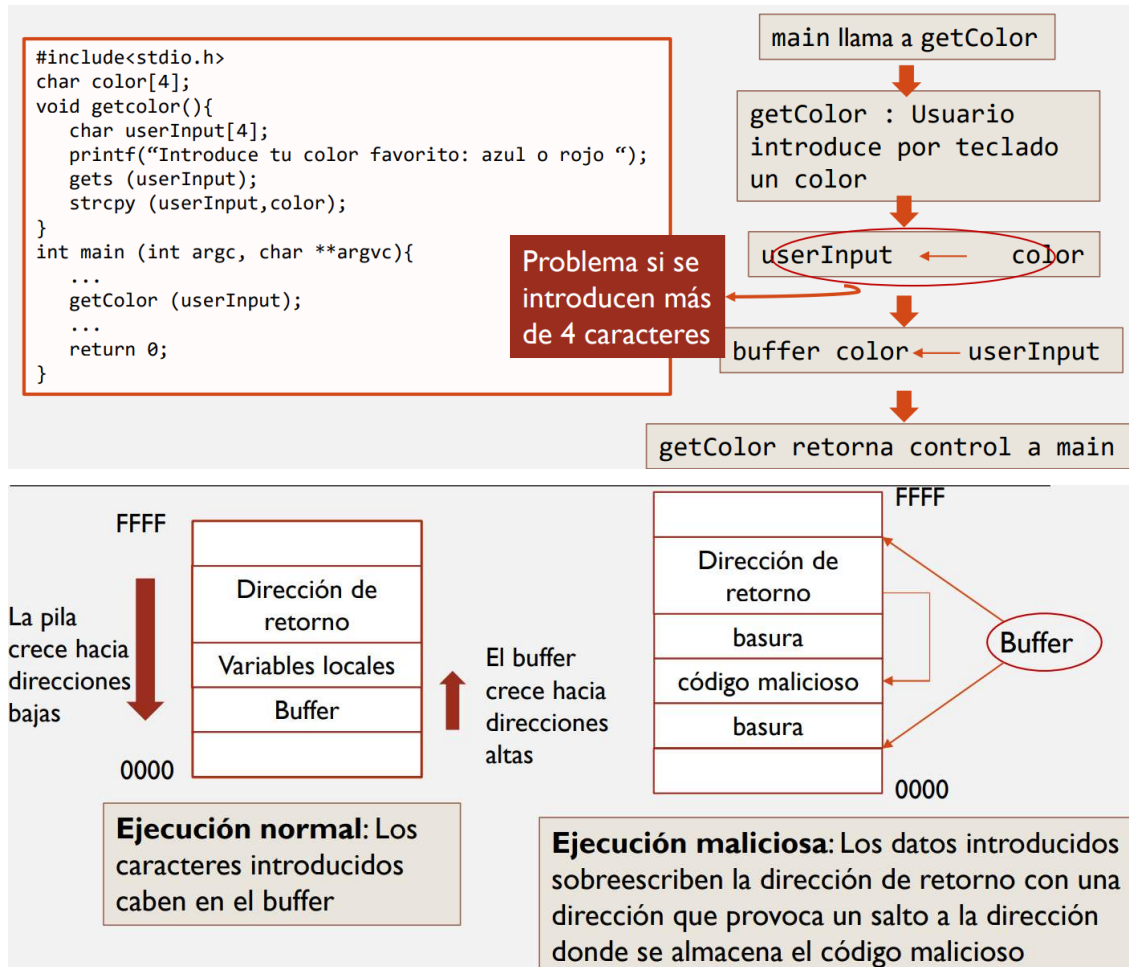
## Memory Overflow

Se produce cuando el programa intenta acceder (lectura o escritura) a una posición de memoria no reservada para esa variable. Ejemplos:

- **Desbordamiento de buffer (Buffer Overflow):** Se produce cuando un programa no controla la cantidad de datos que se copian en un buffer, de forma que si esa cantidad es superior a la capacidad del buffer los bytes sobrantes se almacenan en zonas de memoria adyacentes, sobrescribiendo su contenido original. Ejemplo:



- **Stack Buffer Overflow:** Se produce cuando el buffer que se desborda se encuentra alojado en la pila (stack), puede ser una variable local o un parámetro a una función. Se produce por no chequear la cantidad de datos que se van a introducir en el buffer. Ejemplo:



- **Heap Buffer Overflow:** Es un desbordamiento de buffer que ocurre cuando el buffer está almacenado en el espacio de memoria reservado para el montículo (heap).

```
#define BUFSIZE 256
int main(int argc, char **argv) {
    char *buf;
    buf = (char *)malloc(sizeof(char)*BUFSIZE);
    strcpy(buf, argv[1]);
}
```

Si la cadena de caracteres que se introduce como argumento excede el tamaño reservado para el buffer se produce el desbordamiento

- **Desbordamiento de entero (Integer Overflow):** Se produce cuando se intenta colocar un valor entero dentro de un espacio de almacenamiento que no es lo suficientemente grande como para contener la representación binaria de ese entero.

```
short int bytesRec = 0;
char buf[SOMEBIGNUM];
```

```
while(bytesRec < MAXGET) {
    bytesRec += getFromInput(buf+bytesRec);
}
```



**Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.**  
**No aprobó nada. Lo probó todo.**

- Algunas funciones de C son problemáticas: Permiten el acceso a direcciones bajas de memoria sin comprobar el tamaño de los datos de retorno, pudiendo provocar estados o condiciones no deseadas.

Función	Gravedad	Solución
gets	La más alta	fgets(buf,size,stdin)
strcpy, strcat	Muy alta	strncpy, strncat
sprintf, vsprintf	Muy alta	snprintf, vsnprintf
Familia scanf	Muy alta	Especificadores de precisión
realpath, syslog	Muy alta (dependiendo de la implementación)	Maxpathlen y chequeos manuales
getopt, getopt_long, getpass	Muy alta (dependiendo de la implementación)	Cadenas truncadas de entrada de tamaño razonable

## Malware

Software **diseñado** para dañar o acceder secretamente a un computador sin que el dueño nos haya dado su consentimiento. Engloba distintos tipos de amenazas software (virus, troyanos,...) cuya ejecución implica una acción maliciosa. Los diferentes tipos de malware se suelen combinar para realizar un ataque. Se suelen distribuir a través de (vector de ataque): correo electrónico (compartición de documentos, vídeos, programas,...), descargas de Internet o pueden ser insertados en el sistema por el atacante (entra con credenciales compradas en la Darkweb).

## Características

- Replicación: Si un programa cuenta con esta característica se le considera virus, sin importar que tenga alguna otra característica.
- Métodos de ocultamiento.
- Activación .
- Manifestación: La capacidad que nos hará notar que tenemos código malicioso en nuestro ordenador.
- Explotación y escalada de privilegios.





Un sistema se puede infectar por:

- Correo electrónico.
- Navegación por Internet.
- Documentos o ficheros que aprovechan vulnerabilidades de productos ofimáticos o componentes de navegación por Internet.
- Downloader.
- Waterhole.
- Ataques laterales.




























### Tipos

- **Rootkit:** Conjunto de herramientas diseñadas para ocultar ciertos objetos o actividades del sistema. Se instalan a nivel de kernel, lo que las hace invisibles y les permite un control total.
  - Objetivos: Adquirir máximos privilegios y/o ocultar ficheros o programas.
  - Ejemplos: TDSS, ZeroAccess, Alureon, Necurs.
- **Troyanos:** Son programas maliciosos, camuflados normalmente como software legítimo, que permiten acceso remoto al sistema infectado y realizan acciones no autorizadas por el usuario, como:
  - Objetivos: Utilizar la máquina como parte de una botnet; Instalación de otros programas; Robo de información personal; Borrado, modificación o transferencia de ficheros; Ejecutar o terminar procesos; Apagar o reiniciar el equipo; Monitorizar las pulsaciones del teclado; Realizar capturas de pantalla.
  - Ejemplos: SMS, Spy, Mailfinde, GameThief, PSW, Banker...
- **Virus:** Programa informático diseñado para infectar archivos. Suele venir dentro del código de otros programas, por lo que pueden no ser detectados por los anti-virus. Los virus pueden replicarse y propagarse por otros sistemas.
  - Objetivos: Infectar a otros ficheros o realizar acciones maliciosas.
  - Ejemplos: Ransomware, Boot sector virus, Shell virus, Cluster virus, Multipartite virus, Macro virus...
- **Ransomware:** Virus informático (malware secuestrador) que cifra los archivos de sus víctimas y pide un rescate vía bitcoin para liberar los archivos originales. En el ataque se suele exfiltrar información antes del cifrado y el atacante amenaza con hacer pública esa información si no se paga.
- **Gusanos:** Tipo especial de virus que tiene la capacidad de replicarse así mismo. Es capaz de viajar de una máquina a otra sin necesidad de interacción humana.
  - Objetivo: Colapsar los ordenadores y las redes. Pueden entrar a través de: Email, IM, IRC, Net, P2P...

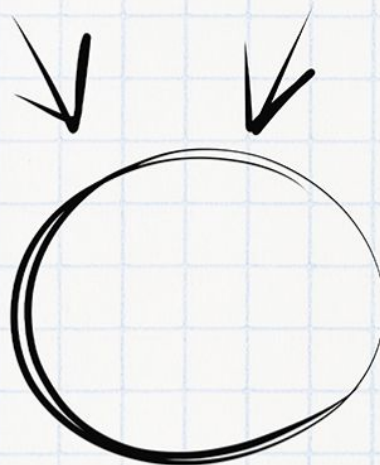
- Ejemplos: Morris Worm, ILOVEYOU, Nimda, Code Red, Melissa, Blaster, Sasser, Storm Worm, Michelangelo, Jerusalem...
- **Bomba lógica:** Parte de código insertada intencionalmente en un programa informático que permanece oculto hasta cumplirse una o más condiciones preprogramadas, en ese momento se ejecuta una acción maliciosa. Ejemplos de acciones: Borrar información del disco duro, Cifrar información del disco duro, Mostrar un mensaje, Enviar un correo electrónico, Apagar el monitor, Apagar el sistema...
- **Botnet:** Equipo infectado con malware que un hacker puede controlar de manera remota. El bot se puede utilizar para ejecutar más ataques o formar parte de una colección de bots llamada botnet que pueden incluir millones de dispositivos que se propagan de forma desapercibida.
- **Puerta trasera (Backdoors):** Secuencia especial dentro de algún software mediante la cual se pueden evitar los sistemas de autenticación para acceder al sistema.
- **Adware:** Programas diseñados para reproducir publicidad en tu ordenador, redirigir tus búsquedas a sitios web publicitarios y recopilar información de marketing sobre tus gustos, como los tipos de sitios web que visitas, de forma que se puedan reproducir anuncios personalizados.
- **Pornware:** Programas que reproducen material pornográfico en un dispositivo. Incluye programas instalados de forma malintencionada, sin conocimiento por parte del usuario, para patrocinar servicios y sitios web de contenido pornográfico sujetos a suscripción.
- **Riskware:** Programas legítimos que pueden provocar daños si son utilizados por ciberdelincuentes, como herramientas de conexión remota, de gestión de contraseña, servidores de descargas...
- **Malware sin archivo:** Software malicioso que utiliza programas legítimos para infectar un equipo. No necesita ningún archivo y no deja rastro, lo que dificulta su detección y eliminación. Las infecciones sin archivos, que no se almacenan en un archivo ni se instalan directamente en una máquina, van directas a la memoria, y el contenido malicioso nunca toca el disco duro.
- **Híbridos:** En la actualidad, la mayoría de los malware son una combinación de diferentes tipos de software maliciosos que suele incluir partes de troyanos, gusanos y, a veces, también un virus.

# Imagínate aprobando el examen

## Necesitas tiempo y concentración

Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios <span>Anual <input type="checkbox"/></span>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,  
¿Qué nota vas a sacar?



# WUOLAH