# Software malicioso

**Curso** 2013/14

**Grado** Gestión Informática Empresarial

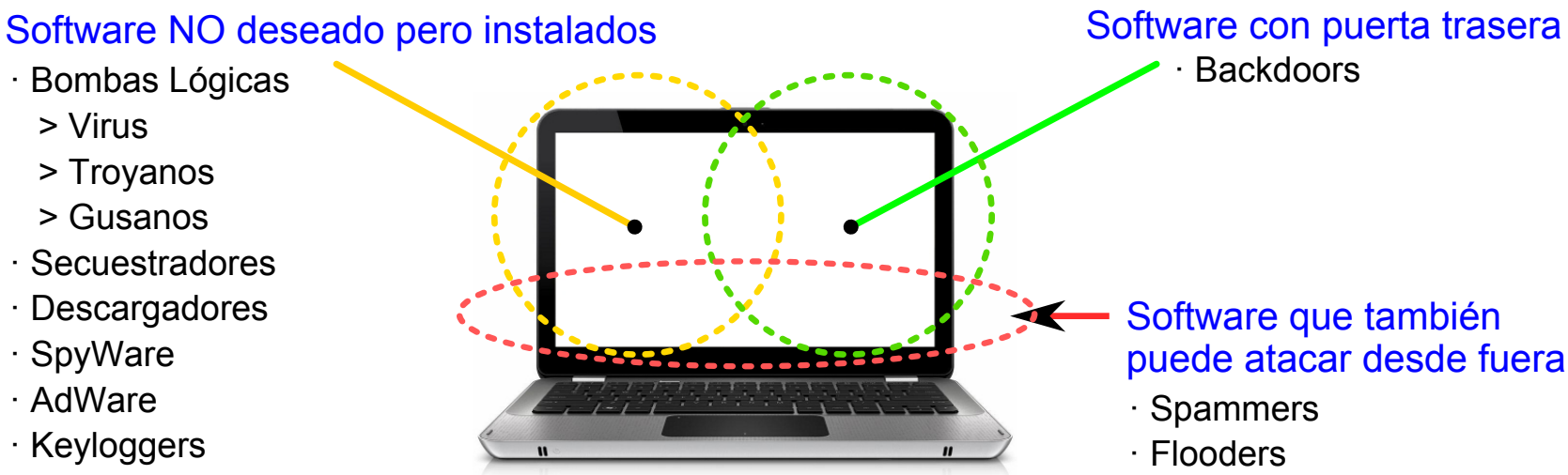
**Asignatura** Auditoría y Seguridad Informática

**Profesores** Alfredo Cuesta Infante
alfredo.cuesta@ajz.ucm.es
Alberto Herrán González
aherran@ajz.ucm.es

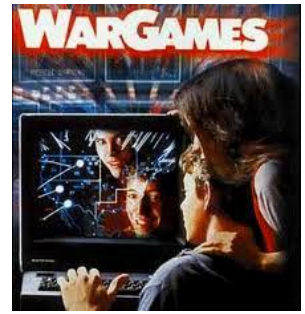
# Contenidos

# Vistazo global

- El software malicioso o *malware* es una amenaza que se presenta como programas que explotan las vulnerabilidades de los sistemas de computación.
- Afecta tanto a aplicaciones como a compiladores, programas del kernel del SO y otras utilidades.
- Los programas pueden estar instalados sin que el usuario sea consciente de ello.
- También pueden ser un software instalado por el usuario pero con funcionalidades desconocidas.
- También podemos ser víctimas de malware instalado en otros sistemas a través de internet.
- Es importante remarcar que las fronteras de estos grupos son *difusas*



# Malware con puerta trasera




## Puerta trasera

- En inglés *backdoor*.
- El usuario instala el sw. por su propia voluntad.
- El sw. tiene funcionalidades extra desconocidas por el usuario.
- La puerta trasera suele ser una palabra clave que se introduce en una contraseña, menú o algún campo solicitado en una instalación.
- Introducido durante la programación o compilación de la aplicación instalada.
- Si lo introduce el programador para depurar código se conoce como *gancho para mantenimiento*, en inglés *maintenance hook*.

## Ejemplo muy básico

```
username = read_username();
password = read_password();
if username == "b4ckd00r"{
    allow_login(admin);
}
if ( valid_user(username) & valid_pass(password) ){
    allow_login(username);
else
    reject_login();
}
```

Se puede ver una secuencia de la película *Juegos de guerra* en la que se explica lo que es un *backdoor* buscando en YouTube 'backdoor war games' y eligiendo el 1º o 2º resultado.

# Malware no deseado pero instalado (1/2)

## Bombas lógicas y sucesores


- **Bomba lógica** *Logic bomb*.
  Código en un programa legítimo que *explota* bajo ciertas condiciones.
- **Troyanos** *Trojan horse*.
  Pieza de sw. (un programa, función, parche,...) de apariencia útil/infonsiva con código malicioso escondido.

**Tipo 1** La función inicial sigue intacta y se ejecuta la otra en paralelo.
**Tipo 2** Modifica la función inicial pero no lo parece.
**Tipo 3** Se reemplaza completamente la función original.

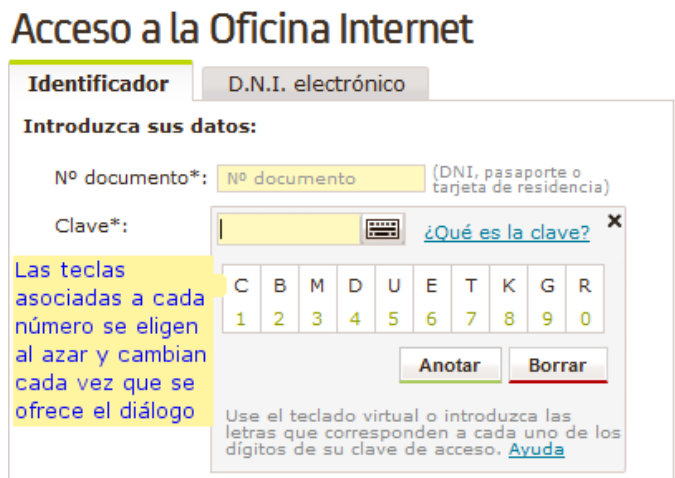
- **Virus** *Virus*.
  Pieza de sw. que se inyecta en otro programa *infectándolo*. El código inyectado además permite la replicación del mismo para infectar otros programas.
- **Gusanos** *Worms*.
  Programa que se puede autoreplicar y envía las copias a través de conexión de red. A diferencia de un virus por e-mail, el gusano busca activamente ordenadores a los que infectar.
- **Secuestradores** *Ramsomware*.
  Programas que cifran archivos importantes para el usuario, haciéndolos inaccesibles y así extorsionar al usuario para poder recibir la contraseña que le permita recuperar sus archivos.

El 26 de septiembre de 1988 la revista TIME dedicó su portada a los virus informáticos.
El artículo se llamaba *The invasion of the data snatchers*.

# Malware no deseado pero instalado (2/2)

## Recolectores de información

- **Espías** *Spyware*.
  Programa que se instala en la computadora con el propósito de recopilar y luego enviar información a un individuo.
- **Capturadores de teclado** *Keyloggers*.
  Un keylogger es un programa que monitoriza todo lo que el usuario teclea y lo almacena para un posterior envío.
  Por ejemplo, un número de tarjeta de crédito puede ser enviado al autor del programa y hacer pagos fraudulentos.
  La mayoría de los keyloggers son usados para recopilar claves de acceso.




## Instaladores

- **Anuncios** *Adware*.
  Son programas que muestran publicidad de forma intrusiva e inesperada, usualmente en forma de ventanas emergentes (pop-ups).
- **Descargadores** *Downloaders*.
  Programa que instala otros elementos en el computador atacado.

# Malware que también puede atacar desde fuera


## Spammers

- Programa para enviar correos electrónicos (principalmente pero no sólo) de forma masiva.
  En 2011 hubo 7 billones de mensajes spam.
- Mínimo porcentaje de éxito / coste de atacar casi nulo = gran beneficio neto.
- El elemento clave es la obtención de direcciones de correo. Hay varias maneras:
  - Grupos de noticias, listas de distribución de correo, blogs.
  - Agendas del propio equipo capturadas, por ej. con un troyano.
  - *Robots* (programas que recorren paginas web) que capturan direcciones de correo.
- **Defensas:** Instalar un filtro anti-spam en el correo electrónico.
  También conviene escribir las direcciones de email de un modo no reconocible por el robot. Por ejemplo la pág.web de GIE, o poniendo una imagen en vez del texto.

Spam es jamon cocido especiado (Spiced Ham) y vendido en latas.
El nombre viene de un *sketch* de los Monty Python en la pelicula *Los caballeros de la mesa cuadrada*. Se puede ver en YouTube buscando 'spam'.

## Flooders

- Programa que provoca un gran volumen de tráfico de red en muy poco tiempo con la intención de saturar una red.
- Es especialmente dañino si pretende lograr una denegación del servicio (*Denial of Service, o DoS*). Por ejemplo el atacante puede envíar un gran número pings (*ping flood*), esperando que el atacado responda, consumiendo ancho de banda tanto de entrada como de salida.
- **Defensas:** Filtrar la IP que ataca en el firewall o grandes cantidades de ICMP.

# Virus

## Infección

Se entiende por **infección** modificar el código de otros programas.
Un virus modifica el código de dos maneras:

- Insertando código malicioso.
- Insertando código para hacer copias de sí mismo. ¡ Pueden ser distintas !

## Estructura y fases

- **Mecanismo de infección** *Infection mechanism*
  El método mediante el cual el virus se replica.
  Cuando empieza a infectar pasa de la fase *letargo* a la fase *propagación*.
- **Gatillo o disparador** *Trigger*
  Suceso que hace *detonar la carga*, i.e. activar el sw. malicioso.
  Este suceso hace pasar a la fase *activación*.
- **Carga** *Payload*
  Lo que el virus hace, además de contagiar. Suele involucrar un daño.
  La carga se *detona* en la fase *ejecución*.

## Clasificación

- **Por blanco:** Según ataquen al sector de arranque, a ficheros o a macros.
- **Por estrategia para camuflarse:**
  1. **Encriptados:** Utilizan claves aleatorias para encriptarse diferente en cada réplica.
  2. **Invisibles:** Comprimen al *anfitrión* para que se reduzca tanto como el tamaño del virus. Así, al adherirse, no se ve diferencia de tamaño respecto del original.
  3. **Polimórficos:** El código del virus *muta* en cada infección pero sigue haciendo lo mismo.
  4. **Metamórficos:** El código se reescribe completamente en cada infección. No sólo cambia su *aspecto*, también su *comportamiento*.

# Ejemplo

**Programa original**

Todo programa, una vez compilado, comienza con una directiva que indica donde está la primera instrucción ejectuable

| *Ir a $X* |
|---|
| *$X* |
| |
| |
| |
| |
| |
| |
| |
| |

**Programa infectado**

| *Ir a $Z* |
|---|
| **firma** |
| **infección** |
| **gatillo** |
| **carga** |
| *$Z* **ejecutar infeccion** |
| **if gatillo == true then detonar_carga** |
| *$X* |
| |
| |
| |
| |
| |
| |
| |
| |

Puede ser simplemente una línea de texto con algo escrito y comentado.

```
while (seguir) {
   fichero ← abrir fichero aleatorio
   if (está la firma del virus)
        seguir ← true ;
   else
        adjuntar_código_virus (fichero)
   }
}
```

Devuelve **true** si se cumple la condición

Lo que sea que haga este virus

# Antivirus

## Desinfección

**Fase 1** **Detección:** Determinar que la infección ha ocurrido y dónde.
**Fase 2** **Identificación:** Averiguar de qué virus se trata.
**Fase 3** **Eliminación:** En el mejor caso supone eliminar el virus restaurando el programa original como estaba antes de la infección. En caso de no ser posible debe ofrecer la posibilidad de *cuarentena* o incluso de eliminar programa+virus.

## Generaciones

1º **Busqueda simple,** de la firma del virus o de una estructura similar.
Suelen mantener una base de datos con los tamaños de cada programa para comparar con el tamaño actual.
2º **Busqueda heurística,** i.e: basada en reglas que tienen alta probabilidad de ocurrir.
- Si es un virus encriptado, buscar la clave al comienzo de la función de encriptación.
- Si el programa incluye algún método de comprobación de integridad, comprobarlos.

3º **Trampas activas,** residentes en memoria, que identifican los virus por su actividad, no por su forma.
4º **Protección total,** con funciones de cada generación anterior.

## ¿Dos mejor que uno?

↓ Si uno sólo hace más lento el sistema, dos peor.
↓ Un antivirus es un programa que recorre todos los archivos husmeando en ellos.
Ese comportamiento es muy similar al de un virus.
Por tanto otro antivirus podría sospechar de él y atacarlo.