

Seguridad Informática

Introducción

Ramón Hermoso y [Matteo Vasirani](#)

Grado en Ingeniería Informática



1 Historia

2 Propiedades

3 Contramedidas

4 Fallos

5 Principios

1 Historia

2 Propiedades

3 Contramedidas

4 Fallos

5 Principios

¿Qué es la seguridad informática?

- **Seguridad** significa **proteger** recursos valiosos, que pertenecen a un legítimo **propietario**, de los posibles **peligros y ataques** perpetrados por agentes **no autorizados**



- La **seguridad informática** se ocupa de proteger los recursos de un sistema informático
 - Información
 - Servicios
 - Arquitecturas

Antes del año 2000

- Hacking como desafío

- Varones por debajo de los 30 años
- Addictos a los ordenadores
- Ningún interés comercial

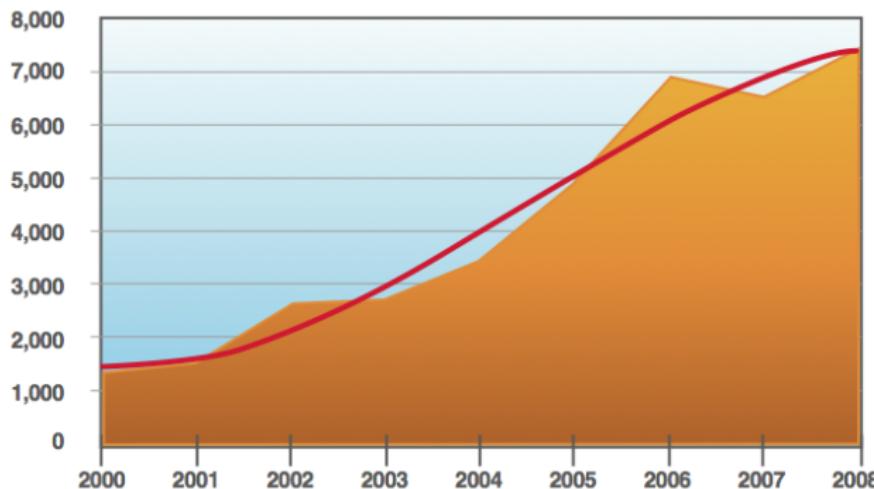


Después del año 2000

- Difusión del acceso a Internet a gran escala
- Hacking por lucro
 - Malware, gusanos, troyanos
 - Robo de información, números de tarjetas de crédito
 - Adware, scareware
 - Publicidad, falsos antivirus
 - Botnet
 - “Alquiler” de la botnet a los spammers
 - Ataques en software lado usuario
 - Navegadores, media-players, lectores PDF
 - Ataques en redes sociales
 - Scam
 - Desbloquear una herencia en Nigeria
 - Phishing

IBM X-Force 2009

Vulnerabilidades software



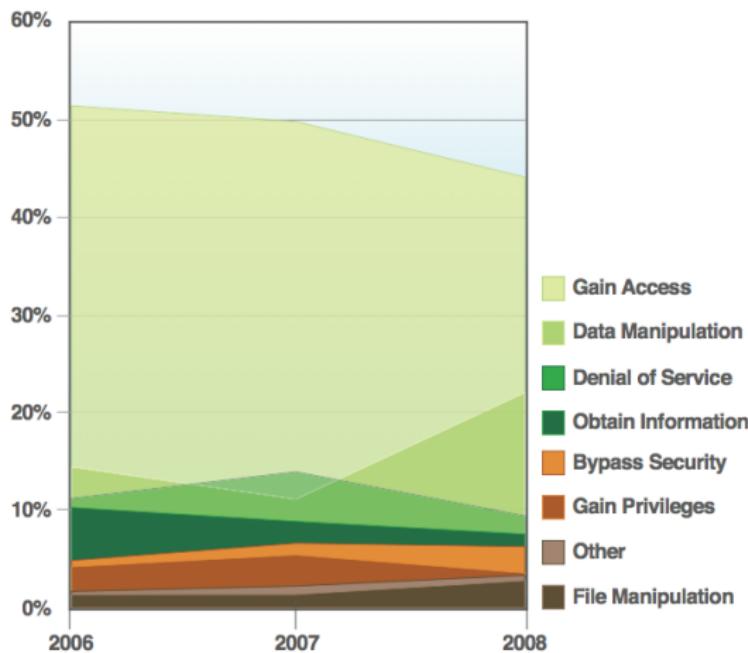
IBM X-Force 2009

Vulnerabilidades software

Ranking	Vendor	Disclosures
1.	Microsoft	3.16%
2.	Apple	3.04%
3.	Sun	2.19%
4.	Joomla!	2.07%
5.	IBM	2.00%
6.	Oracle	1.65%
7.	Mozilla	1.43%
8.	Drupal	1.42%
9.	Cisco	1.23%
10.	TYPO3	1.23%

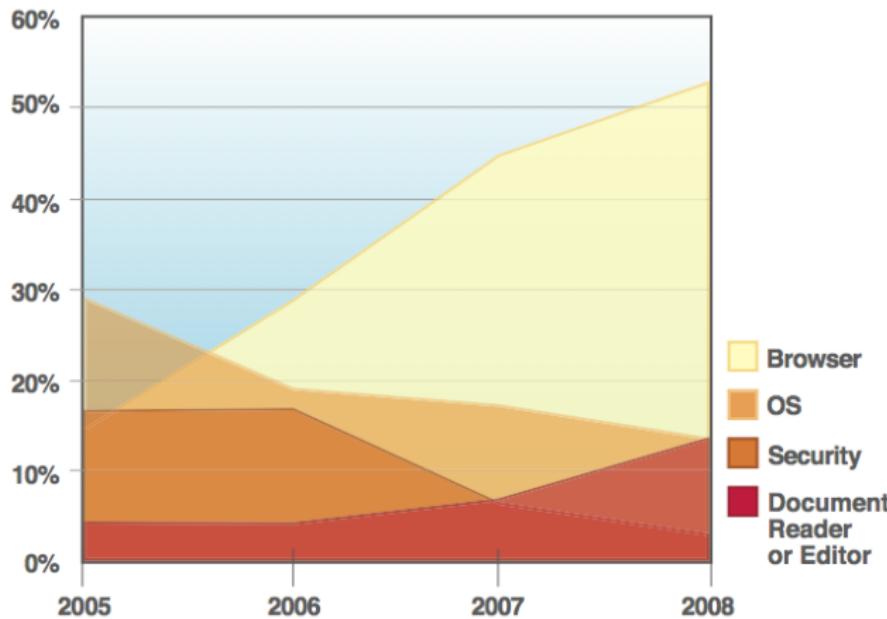
IBM X-Force 2009

Tipo de ataque



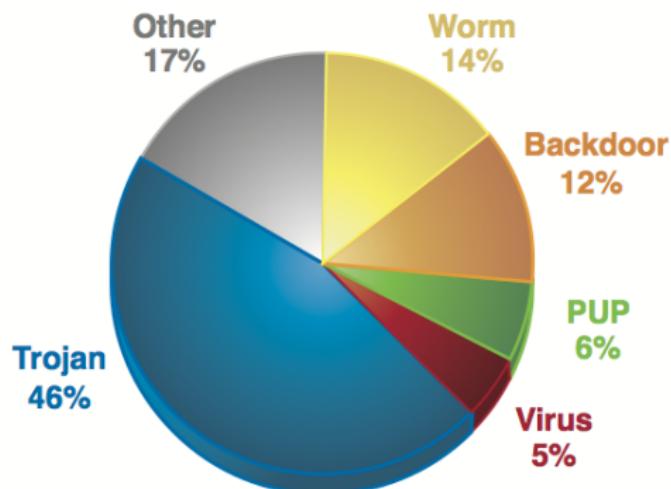
IBM X-Force 2009

Tipo de programa atacado



IBM X-Force 2009

Malware



¿Por qué tantas vulnerabilidades?

- ¿ Por qué se desarrolla tanto software inseguro?
 - Poca énfasis en la programación segura en las universidades y en los libros de programación
 - Lenguajes no seguros (p.e., C, Ada, Pascal)
 - Incentivos económicos para librar software rápidamente y luego parchearlo
 - Difícil detectar fallos (p.e., kernel Linux 2.4)

¿Por qué tantas vulnerabilidades?

- ¿ Por qué se desarrolla tanto software inseguro?
 - Poca énfasis en la programación segura en las universidades y en los libros de programación
 - Lenguajes no seguros (p.e., C, Ada, Pascal)
 - Incentivos económicos para librar software rápidamente y luego parchearlo
 - Difícil detectar fallos (p.e., kernel Linux 2.4)

```
if ((options == (_WCLONE | _WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

¿Por qué tantas vulnerabilidades?

- ¿ Por qué se desarrolla tanto software inseguro?
 - Poca énfasis en la programación segura en las universidades y en los libros de programación
 - Lenguajes no seguros (p.e., C, Ada, Pascal)
 - Incentivos económicos para librar software rápidamente y luego parchearlo
 - Difícil detectar fallos (p.e., kernel Linux 2.4)

```
if ((options == (_WCLONE | _WALL)) && (current->uid = 0))
    retval = -EINVAL;
```

```
if ((options == (_WCLONE | _WALL)) && (current->uid == 0))
    retval = -EINVAL;
```

1 Historia

2 Propiedades

3 Contramedidas

4 Fallos

5 Principios

Propiedades que hay que asegurar

● Confidencialidad

- Hay información que es **privada** y no se quiere compartir con personas no autorizadas
 - Número de tarjeta de crédito
 - Listado de empleados y pagas
 - Histórial médico
 - Fotos de aquella fiesta de cumpleaños
- La confidencialidad supone la noción de agente **autorizado** y de una **política de seguridad** que diga quién puede o no puede acceder a la información.

Propiedades que hay que asegurar

● Integridad

- Se quiere que la información no sea **alterada** (de manera malintencionada) por personas no autorizadas
 - Programa infectado por un virus
 - Datos de una transferencia bancaria
- La integridad se puede caracterizar como la **escritura no autorizada** de la información.
- Se supone la existencia de una política de seguridad que diga quién puede o no puede escribir y modificar la información

Propiedades que hay que asegurar

● Disponibilidad

- Se debe poder acceder a la información y a los servicios **cuando se desea** y de la **modalidad esperada**
 - Denegación de servicio (DoS)
 - Adware
- Hay que saber distinguir un ataque de un uso legitimo del servicio.

Propiedades que hay que asegurar

● Autenticación

- Se debe poder verificar de manera precisa la **identidad** de los usuarios y la origen de los datos
 - Romper llaves de seguridad
 - Robo de identidad
- La autenticación es un **prerequisito** de cualquier política de seguridad que quiera autorizar o denegar el acceso a la información.
- Los métodos de autenticación de un usuario se basan en
 - **algo que posee** (p.e., smart-card)
 - **algo que sabe** (p.e., contraseña)
 - **algo que lo caracteriza** (p.e., huella)

1 Historia

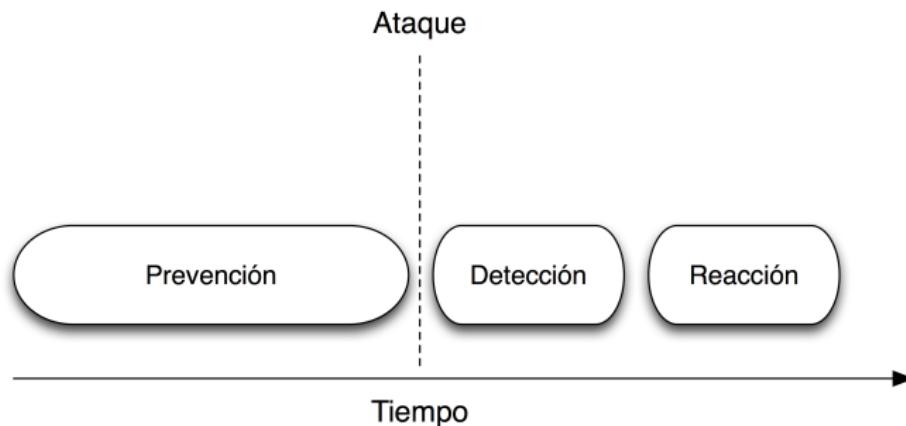
2 Propiedades

3 Contramedidas

4 Fallos

5 Principios

Contramedidas de seguridad



Contramedidas de seguridad

● Prevención

- Evitar que se pueda comprometer una o más de las propiedades que hay que asegurar.
- Diseño seguro del sistema
- Tecnologías de prevención
 - Confidencialidad → Cifrado
 - Integridad → Permisos de acceso
 - Disponibilidad → Filtrar tráfico
 - Autenticación → Certificado digital
- La prevención es la contramedida más importante

Contramedidas de seguridad

● Detección

- Si alguien intenta comprometer una de las propiedades, hay que asegurarse de detectar dicho intento, para poder contrarrestarlo o restablecer el sistema
- Muy importante en seguridad informática, ya que la detección de un “robo de datos” no implica la falta de dichos datos.
- Tecnologías de detección
 - Logs
 - Checksum (código hash para detectar alteración de ficheros)
 - Intrusion detection system (IDS)

Contramedidas de seguridad

● Reacción

- Si se ha detectado un fallo de seguridad, hay que tener un plan para corregir el fallo o restablecer la integridad el sistema
 - Backups
 - Informar las partes implicadas
 - Informar las autoridades

1 Historia

2 Propiedades

3 Contramedidas

4 Fallos

5 Principios

¿Que puede fallar?

- Fuerzas opuestas actúan contra el sistema
 - Complejidad
 - La experiencia y las habilidades de los atacantes
 - Factores humanos impredecibles

¿Que puede fallar?

- Fuerzas opuestas actúan contra el sistema
 - Complejidad
 - La experiencia y las habilidades de los atacantes
 - Factores humanos impredecibles

Rick Cook, *The Wizardry Compiled*

Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to build bigger and better idiots. So far, the universe is winning.

¿Que puede fallar?

● Complejidad

- La complejidad es enemiga de la seguridad, ya que genera errores y fallos (KISS: keep it simple, stupid!)
- \oplus complejidad $\rightarrow \oplus$ bugs $\rightarrow \ominus$ seguridad
- Dos posibles enfoques:
 - ① **Limitar la complejidad**, usando patrones de diseño comprobados y protocolos abiertos
 - ② **Combatir la complejidad**, implementando niveles de seguridad múltiples e invirtiendo más en la detección que en la prevención

¿Que puede fallar?

● Factores humanos

- Los usuarios son increíblemente hábiles en poner en peligro un sistema informático
 - **Procesos:** contraseñas débiles, ignorar advertencias de peligro, desabilitar controles de seguridad
 - **Ingeniería social:** proporcionar información de manera accidental (o a cambio de dinero)
 - **Falta de conocimiento de los riesgos:** abrir ficheros adjuntos llamados virus.exe

¿Que puede fallar?

● Atacantes

- Los atacantes tienen muchas motivaciones para actuar malintencionadamente (p.e., robo, fraude, vandalismo, terrorismo, juego, exhibicionismo)
- Diferentes **perfíles** (experiencia, recursos, dedicación)
 - Script kiddie (bajan script que explotan fallos conocidos sin saber qué hacen)
 - Hacker por hobby (escriben sus scripts, encuentran nuevos fallos)
 - Hacker determinado (cyber-criminales, ataques dirigidos)
 - Hacker profesional (espionaje industrial, mucho conocimiento, muchos recursos)
 - Experto de servicios de seguridad (seguridad nacional, espionaje internacional, soporte gubernamental y legal)
- Es necesario estimar el perfil del atacante en fase de **análisis de riesgo** para poder determinar qué nivel de protección se necesita

1 Historia

2 Propiedades

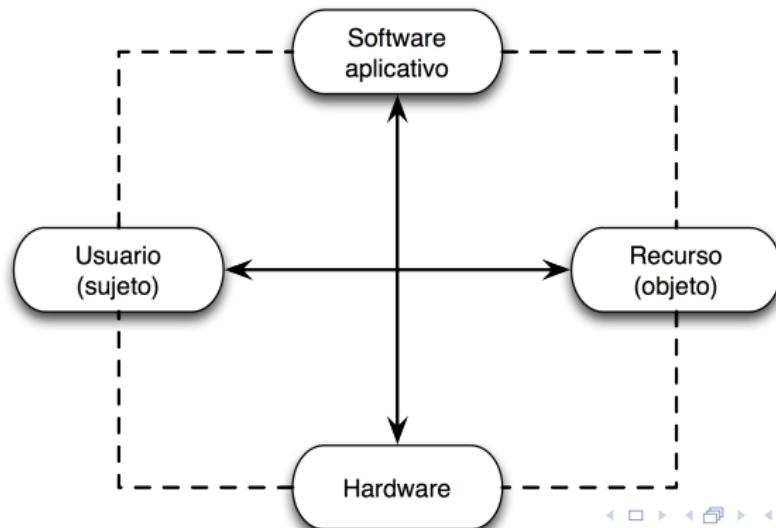
3 Contramedidas

4 Fallos

5 Principios

Principios de seguridad informática

- **Las dimensiones de la seguridad informática**
 - La **política de seguridad** se puede centrar en los **usuarios** o en los **recursos**
 - Los **mecanismos de protección** que implementan la política de seguridad se pueden aplicar desde en nivel hardware hasta el nivel software aplicativo.



Principios de seguridad informática

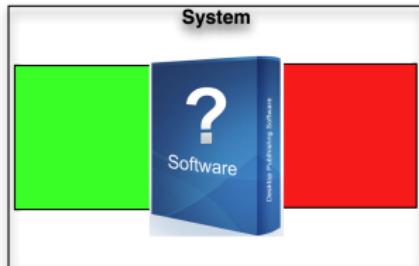
1^a decisión de diseño

¿Cuáles son las partes del sistema que necesitan mecanismos de protección?

Principios de seguridad informática

1^a decisión de diseño

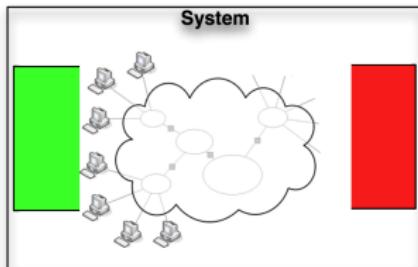
¿Cuáles son las partes del sistema que necesitan mecanismos de protección?



Principios de seguridad informática

1^a decisión de diseño

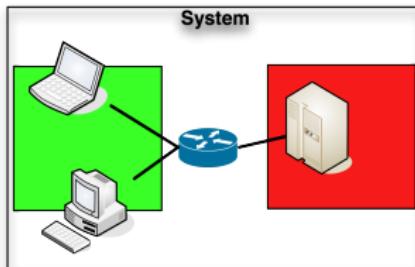
¿Cuáles son las partes del sistema que necesitan mecanismos de protección?



Principios de seguridad informática

1^a decisión de diseño

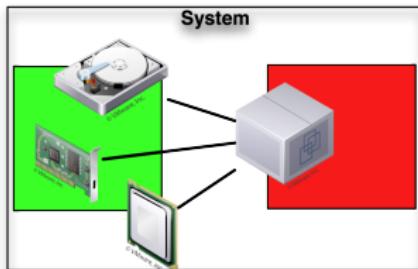
¿Cuáles son las partes del sistema que necesitan mecanismos de protección?



Principios de seguridad informática

1^a decisión de diseño

¿Cuáles son las partes del sistema que necesitan mecanismos de protección?



Principios de seguridad informática

2^a decisión de diseño

¿La política de seguridad tiene que centrarse en los datos, en las operaciones o en los usuarios?

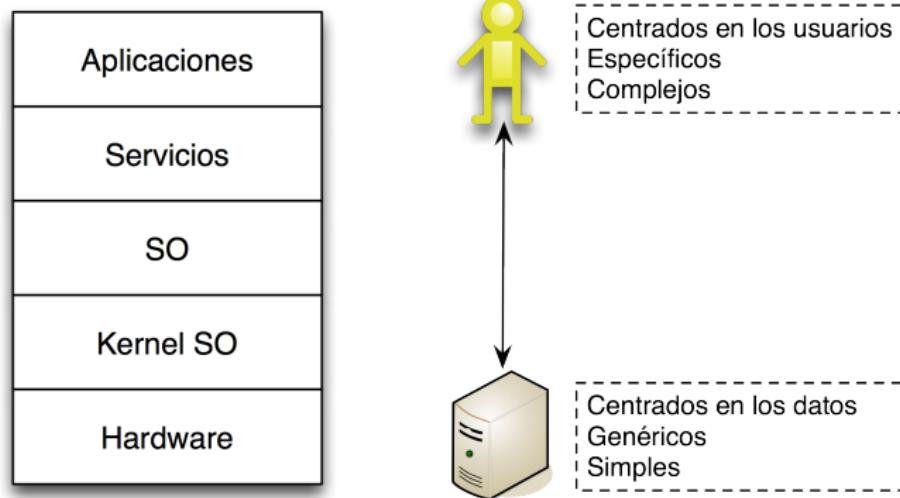
- Ejemplos:

- Datos: una regla prescribe que el campo `balance` de una cuenta corriente tiene que ser un entero
- Operaciones: una regla prescribe que sólo las operaciones `abrir_cuenta()`, `depositar()` y `retirar()` pueden acceder al campo `balance`
- Usuarios: una regla prescribe que sólo el propietario de la cuenta y los empleados del banco pueden acceder al campo `balance`

Principios de seguridad informática

3^a decisión de diseño

¿A qué nivel se implementan los mecanismos de protección?



Resto de la asignatura

- Autenticación y control de acceso
- Seguridad software
- Criptografía
- Seguridad en redes y web