Curso: (62612) Diseño de aplicaciones seguras

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Tema IV: Principios básicos

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Principios básicos

- La seguridad de los programas es compleja
- Utilizar 'listas de la compra' para saber qué evitar o dónde mirar no es siempre una buena idea
- Protegerse contra ataques desconocidos es mucho más complicado que defenderse de problemas conocidos
- Buenos principios: nos defienden no sólo de lo conocido, también de 'lo que puede venir'
- Evitaremos muchos problemas (pero no todos!) siguiendo estos principios

Los principios

1. Asegurar el eslabón más débil
2. Seguridad en profundidad
3. Fallar de modo seguro
4. Seguir el principio del mínimo privilegio
5. Compartimentalizar
6. Simplicidad
7. Promover la privacidad (intimidad)
8. Recordar: guardar secretos es difícil
9. Ser desconfiados
10. Utilizar los recursos de tu comunidad

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Asegurar el eslabón más débil

La seguridad es como una cadena . . .

- Los atacantes buscarán el eslabón más débil
- Incluso aunque busquen en todos los sitios, es más fácil que encuentren problemas allí
- Hay mucho más dinero en un banco que en una tienda normal, pero ¿quién es más fácil que sufra un robo?
- La criptografía raramente es la parte débil: un atacante intentará 'romper' nuestro sistema en otro sitio

Asegurar el eslabón más débil

- La identificación del punto más débil tiene que ver con el análisis de riesgos
- Si tenemos un buen análisis, mitigar el riesgo más grave primero es mejor que mitigar el riesgo más 'sencillo'
- No siempre el programa es el eslabón más débil; a veces tiene que ver con lo que le rodea.

http://www.flickr.com/photos/bulle_de/349393319/

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Asegurar el eslabón más débil

- De nada sirve un programa muy seguro si el acceso físico al servidor es fácil
- 'Ingeniería social': teléfono, paciencia, confianza ...
  - Limitar las capacidades del personal técnico tanto como sea posible
    Ejemplo: cambiar la clave

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Defensa en profundidad

- Estrategias defensivas diversas, en niveles
- Si algo falla, hay más barreras detrás
  - ... pensar en un banco
- Cuidado, no confundir con lo de la cadena ... se trata de protección redundante

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Fallar de forma segura

- Cualquier sistema suficientemente complejo fallará
- Hay que procurar que eso no suponga problemas de seguridad
- Tarjetas de crédito ...
  1. Llamada a la empresa
  2. Comprobación denuncia de robo
  3. Analísis de la compra ....

  Y las bacaladeras? Seguridad vs Conveniencia!

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Fallar de forma segura

- Cualquier sistema suficientemente complejo fallará
- Hay que procurar que eso no suponga problemas de seguridad
- Tarjetas de crédito ...
  1. Llamada a la empresa
  2. Comprobación denuncia de robo
  3. Analísis de la compra ....

  Y las bacaladeras? Seguridad vs Conveniencia!

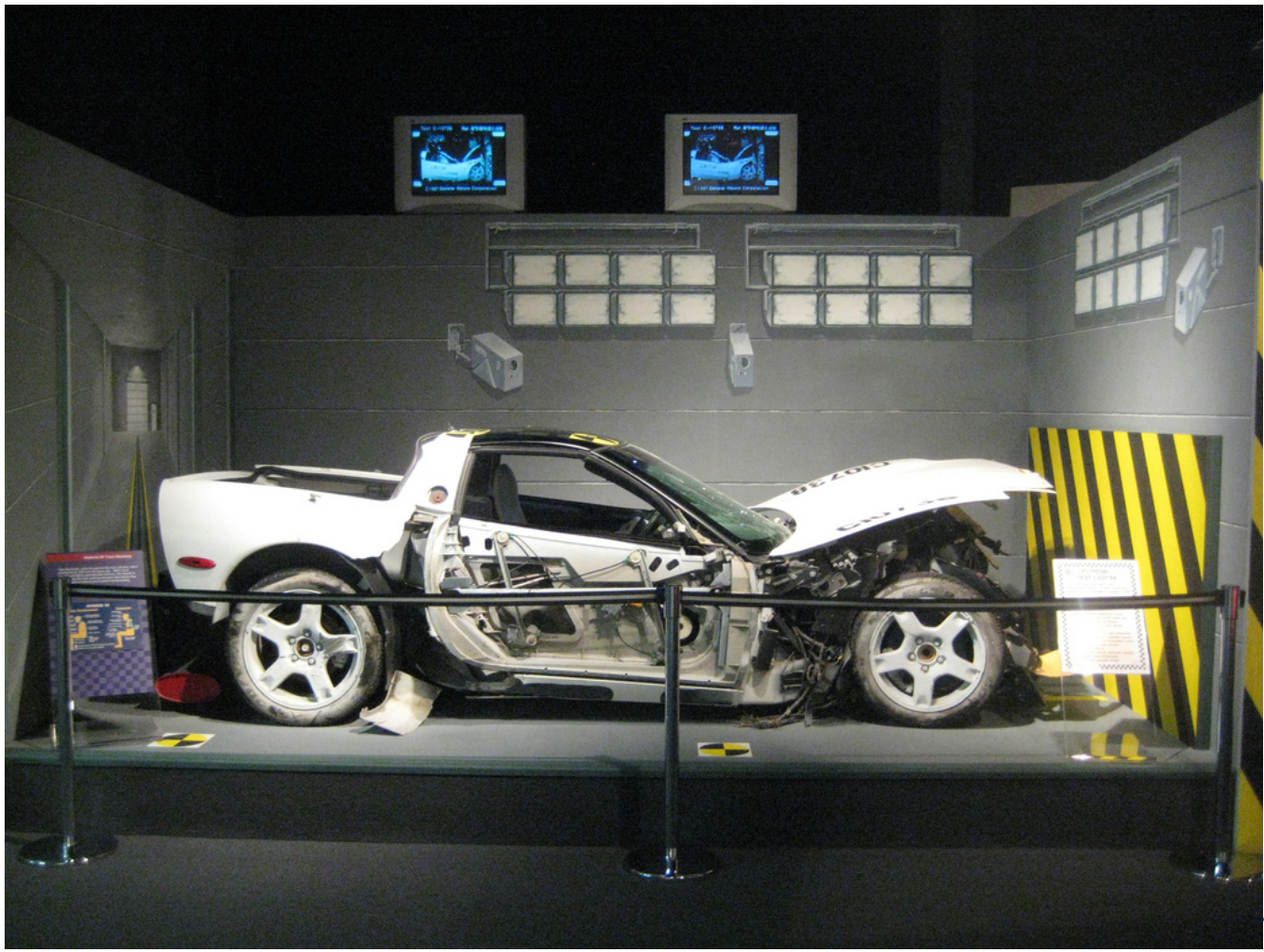
Acciones por defecto seguras

- Si se rompe un sistema de control
  - ... de una caja fuerte, ¿cómo debería quedar?
  - ¿y de un cine?
  - ¿y en un sitio web controlado por claves?

Fallar de forma segura

Mejor aún: degradación controlada

- Cuando un coche choca, hay partes que se deforman.



http://www.flickr.com/photos/bargas/3695903512/

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Fallar de forma segura. ¿Y las configuraciones?

- Por defecto, el sistema lo mas seguro posible
- Recordad que los ejemplos terminan usándose
- Avisar de las consecuencias de los cambios

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Principio del mínimo privilegio

- Dar el acceso mínimo necesario para la tarea encargada
- Mínimo tiempo posible
  - Si alguien nos recoje el correo durante las vacaciones ... ¿le damos también las llaves del piso?
  - Cuando compramos un piso, ¿conservamos la cerradura?
- Ventanas de vulnerabilidad (*windows of vulnerabity*) tan pequeñas como sea posible

Principio del mínimo privilegio

- Si necesitamos permiso para leer un objeto, no darlo también para escribir
- Una vez que hayamos leido, quitarse los privilegios, si no hacen falta más
- La pereza es nuestro enemigo

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Compartimentalización

- Si además del correo, me piden que le dé de comer al perro ¿tengo un corral dónde pueda estar el animal?
- Se trata de minimizar el daño que pueden hacernos (submarinos, petroleros, prisiones, ...)
- El sistema de permisos de Unix es un buen ejemplo de como **no** hacer las cosas
- En la mayoría de las plataformas no está previsto defender una parte del sistema del resto

Simplicidad

- Kiss It Simple Stupid!: La complejidad incrementa la posibilidad de que aparezcan problemas
- El diseño y la implementación debería ser tan sencillo como se pueda
  - Diseños complejos son más difíciles de entender
  - Más difícil de mantener
  - Más fallos

Simplicidad

- Reutilizar cuando sea posible
- Sobre todo para la criptografía
- ¿Contradicción con la defensa en profundidad?
  ... compromiso!!!!
- No es trivial
- Idea: Concentrar las operaciones críticas de seguridad en pocos puntos.
- Sin atajos!!

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Simplicidad

- Ergonomía (*usability*)
- Todos los usuarios deberían tener acceso a la mejor seguridad de nuestro sistema, y no poder introducir puntos inseguros por descuido.
  Algunas ideas (Norman 1989, Nielsen 1993):
  1. Los usuarios no leen
  2. Hablar con los usuarios
  3. Los usuarios no tienen razón siempre
  4. Los usuarios son perezosos

Simplicidad

Making the simple complicated is commonplace; making the complicated simple, awesomely simple, that's creativity.

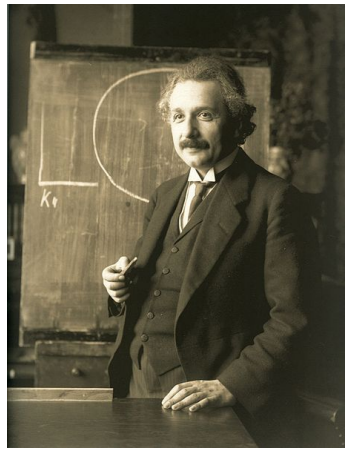
Charles Mingus

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Simplicidad

Making the simple complicated is commonplace; making the complicated simple, awesomely simple, that's creativity.

Charles Mingus



Make everything as simple as possible, but not simpler.

Albert Einstein

(Fotos: Wikipedia)

Promover la privacidad

- Muchos usuarios consideran su intimidad un asunto de seguridad
- Tenemos que tratar de no comprometer los datos de nuestros usuarios
- La mayoría de sitios 'serios' no guarda nuestra clave, ni el número de la tarjeta, ...
  - Al menos, no mostrarla (nunca entera)
  - Almacenarla cifrada
  - Almacenarla en otra máquina diferente

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Promover la privacidad

No sólo los usuarios (¿y el servidor?)

```
telnet pardillo.donde.estas
Trying 1.2.3.4...
Connected to pardillo.donde.estas
Escape character is '^]'.

SunOS 5.7
```

- ¿Hace falta el servicio?
- Si hace falta, ¿es necesario decir el sistema?
- Cuidado!

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Esconder secretos es difícil

- No divulgar datos de los usuarios
- No divulgar claves
- Los 'malos' son muy hábiles (DVD, chips consolas, iPhone, ...)
- Ataques desde dentro
  - Descontentos
  - Inadvertidos
  - En todo caso ... la mayoría

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Cuidado con la confianza

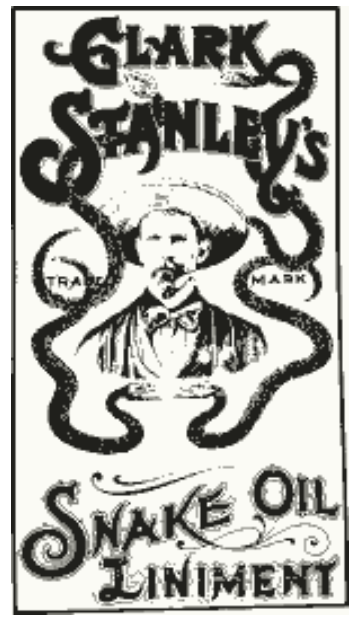
- No poner secretos en el código del cliente
- Diseñar los servidores para no confiar en los clientes
- Ser desconfiado puede ayudar en la compartimentalización

Muchos desarrolladores, arquitectos y gestores de proyectos no saben mucho sobre diseño seguro (incluso los que hacen herramientas de seguridad).

Cuidado con la confianza

- No extender la confianza al servicio de atención al cliente (ingeniería social)
- Seguir a la manada: a veces se hacen algunas cosas por influencia de los competidores
- El escepticismo es sano (cuidado con el aceite de serpiente)

http://www.interhack.net/people/cmcurtin/snake-oil-faq.html



http://en.wikipedia.org/wiki/File:Snake-oil.png

- Es sano desconfiar hasta de uno mismo
- Confianza transitiva (amigos de amigos...)

La confianza

- 'Fronteras' de la confianza
  ¿En quién o en qué confiamos? ¿Qué hacemos cuando pasan datos de una parte menos confiable a una confiable?
- La cadena de confianza
  Confío en ... Que confía en ... Que confía en ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Otros principios

- Exactidud, precisión ('accuracy')
  ¿Corresponde el diseño (y el software luego) a la 'realidad'?
- Claridad
- Acoplamiento débil
  Menos complejidad, menos dependencias, ...
- Cohesión fuerte
  Los módulos gestionan tareas relacionadas entre si: hay una buena descomposición y modularización.
- Adecuada gestión de fallos

La comunidad

- No seguir ciegamente a la masa pero ...
  - El uso continuado de una tecnología ayuda
  - El escrutinio público también

  Ejemplo: la criptografía, bibliotecas seguras, ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Los principios de Microsoft

$$SD^3 + C$$

- Seguro por diseño
- Seguro por defecto
- Seguro en la implantación ('Deployment')
- Comunicación (con los clientes)

`http://msdn.microsoft.com/en-us/magazine/cc163705.aspx`

Atención a ...

- Flujo de los datos de entrada
- Relaciones de confianza (transitividad, ...)
- Suposiciones y Confianza inadecuadamente depositada
- Interfaces
  Los programas se comunican con otros y con el exterior.
- Ataques relacionados con el entorno
- Ataques relacionados con la gestión de fallos
- Condiciones excepcionales
- Configuraciones por defecto que son defectuosas
- Servicios innecesarios