Seguridad en Tecnologías de la Información

Tema 1
Seguridad en las Tecnologías de la Información. Amenazas de Seguridad

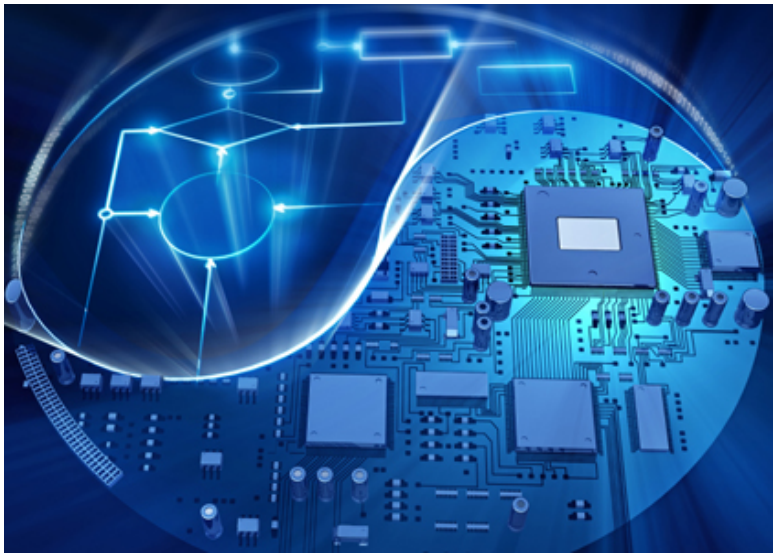
Manuel J. Lucena López
mlucena@ujaen.es

Departamento de Informática
Universidad de Jaén

11 de septiembre de 2024

Las Tecnologías de la Información

- Llevan bastante tiempo entre nosotros.
- Su evolución está siendo cada vez más rápida.
- Sirven para *extender* nuestra memoria.
- Y nuestra capacidad de comunicación.
- Para que resulten útiles, tienen que cumplir una serie de requisitos.



Índice

Concepto de seguridad

En general, un sistema se considera seguro cuando *actúa como debe*.

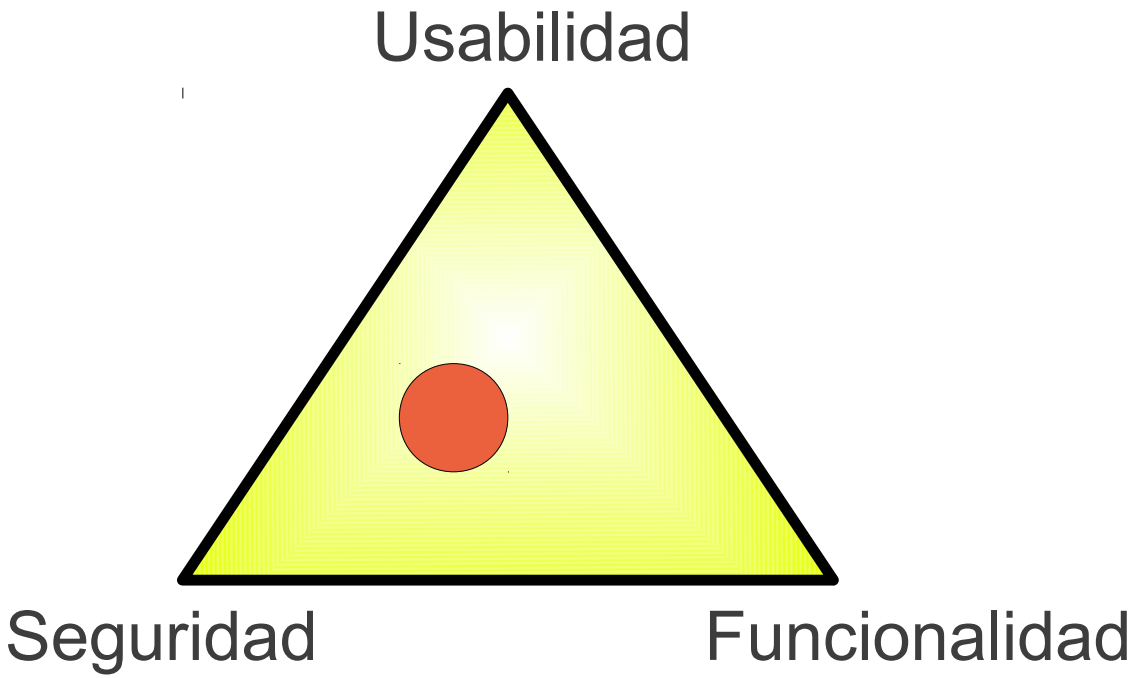
- Un sistema *seguro* genera confianza. Cuanto más seguro, más confiable y viceversa.
- A mayor complejidad, mayor probabilidad de tener fallos.
- Todos los sistemas de información tienen fallos que pueden...
  - no afectar a la información.
  - dañar la información por sí mismos.
  - pasar desapercibidos, pero son susceptibles de ser aprovechados por terceros para provocar daños al sistema.

Sistema de información seguro

Un sistema de información se considera seguro si...

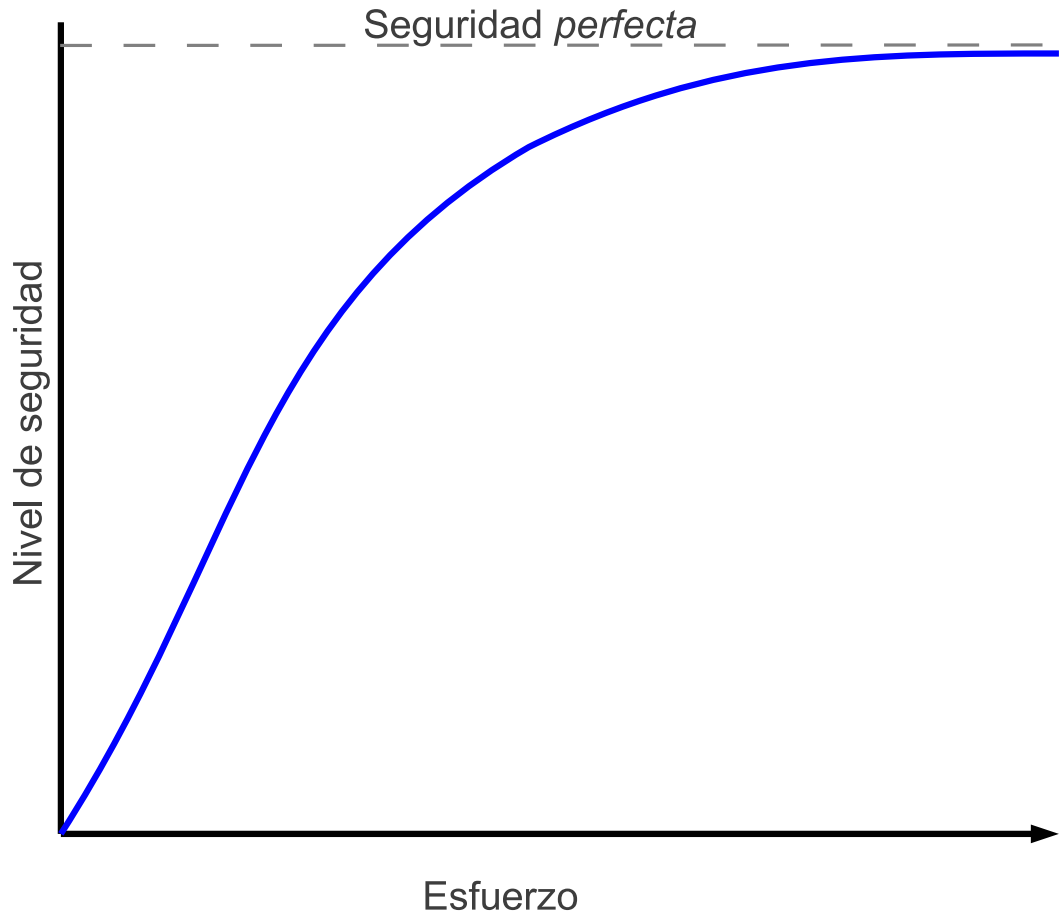
- cuando se produce algún tipo de funcionamiento anómalo, no afecta a la información almacenada o, en su caso, la misma puede ser recuperada en un tiempo razonable.
- la probabilidad de que, de manera deliberada e inadvertida, se produzca un robo, manipulación, o interrupción del servicio, es nula o está por debajo de un límite tolerable.

Seguridad, usabilidad y funcionalidad



- Cuando desarrollamos *software*, buscamos estas tres características.
- Mejorar cualquiera de las tres implica degradar las otras dos.

Esfuerzo dedicado a la seguridad



Hay que encontrar un compromiso entre el esfuerzo (coste) del sistema y su nivel de seguridad.

Propiedades de un sistema de información seguro

Confidencialidad:

Solo pueden acceder a la información aquellos agentes que están autorizados para ello.

Integridad:

La información no sufre alteraciones cuando se almacena, recupera o transmite.

Disponibilidad:

La información puede ser utilizada siempre que se necesite.

Daño, ataque y riesgo

Daño:

Perjuicio que se produce a raíz de un fallo en un sistema.

- Económico, físico, moral, etc.
- Fortuito o provocado.

Ataque:

Acción de provocar un daño a un sistema de forma intencionada.

Riesgo:

Producto entre la magnitud del daño (d) y la probabilidad de que éste ocurra p_d:

$$R = d \cdot p_d$$

Un daño bajo pero muy probable puede suponer más riesgo que un daño mayor, aunque muy poco probable.

Amenaza, vulnerabilidad y *exploit*

Amenaza:

Situación de daño cuyo riesgo de producirse es significativo.

Vulnerabilidad:

Deficiencia de un sistema susceptible de producir –accidental o intencionadamente– un fallo en el mismo.

Exploit:

Técnica que permite aprovechar una vulnerabilidad para producir un daño, y romper la seguridad de un sistema.

Identificación unívoca de vulnerabilidades

CVE: *Common Vulnerabilities and Exposures*

- Proporcionan un identificador *universal* para las vulnerabilidades.
- Elementos:
  - Un identificador con el formato CVE-AAAA-NNNN.
    - AAAA es el año y NNNN es un número único de 4 o más dígitos.
  - Estado: candidato *(candidate)* o definitivo *(entry)*.
  - Breve descripción de la vulnerabilidad.
  - Referencias.

Valoración del impacto de una vulnerabilidad

CVSS: *Common Vulnerability Scoring System*

- Permite medir la peligrosidad de una vulnerabilidad.
- Combina tres aspectos:
  - Base: intrínsecos a la vulnerabilidad (acceso local o remoto, impacto en integridad, disponibilidad y confidencialidad...).
  - Temporal: evolución de la vulnerabilidad (explotabilidad, existencia de contramedidas...).
  - Del entorno: relativos a una implementación concreta y los elementos que la rodean.
- Devuelve un valor de impacto entre 0 (ninguno), 0.1-3.9 (bajo), 4-6.9 (medio), 7-8.9 (alto) y 9-10 (crítico).
- Última versión: 4.0 (junio 2022).
  - Mayor granularidad, menos ambigüedad, aplicabilidad a entornos de salud, seguridad para personas y entornos de control industrial.

Causas y detección de vulnerabilidades

Causas:

- Mal diseño del sistema.
- Implementación deficiente:
  - A nivel de *software*: *Bugs*.
  - A nivel de *hardware*.
- Un uso inadecuado:
  - Un sistema puede enfrentarse a situaciones de uso para las que no está diseñado, dando lugar a fallos y vulnerabilidades.
  - Esta circunstancia suele ser bastante frecuente.

Detección:

- Estudiando el sistema en las fases de diseño, implementación y uso.
- Aprendiendo de ataques externos, a través de sistemas trampa *(Honeypots)*.

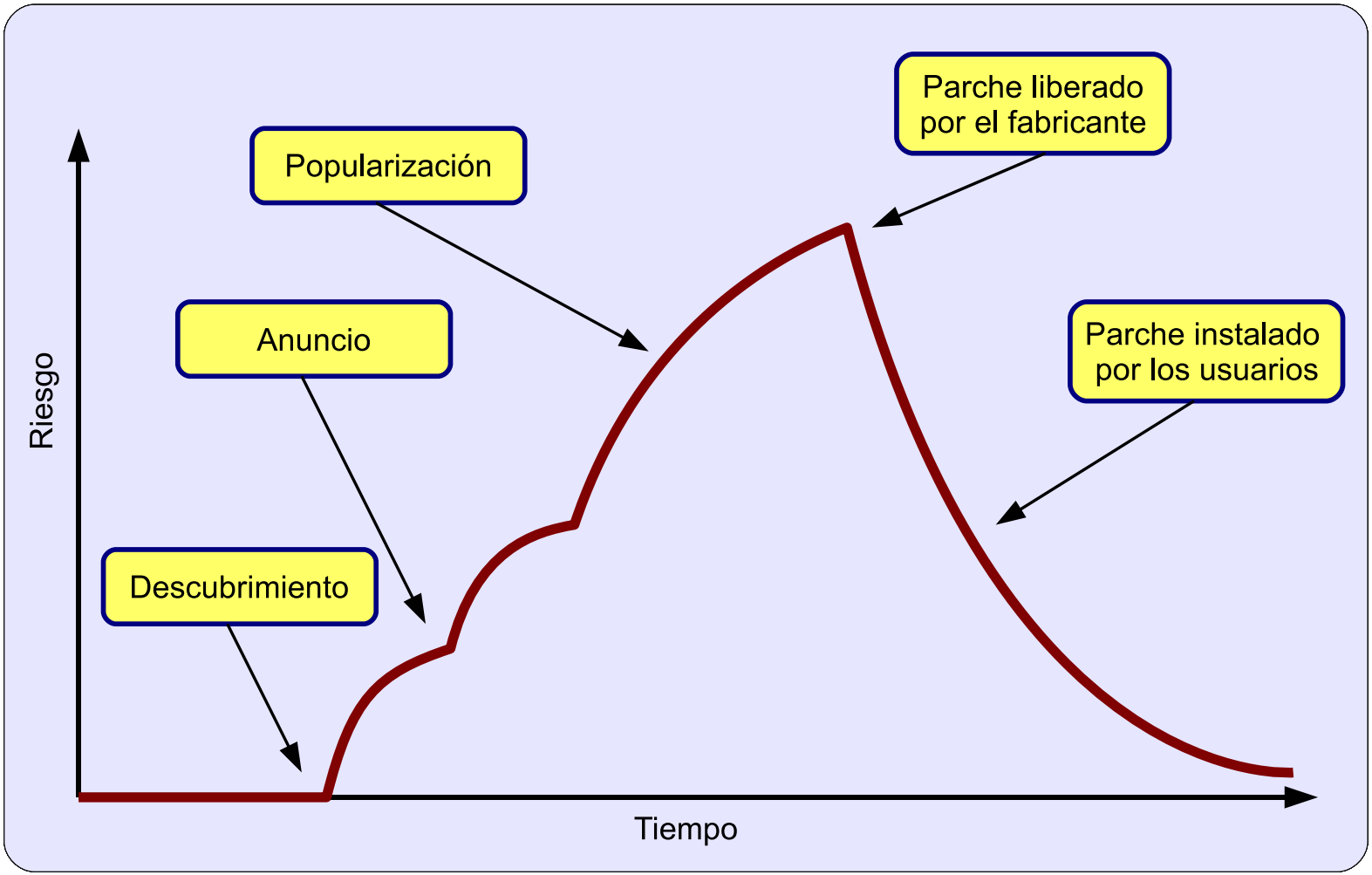
Ventana de exposición

- Tiempo que transcurre desde que se detecta una vulnerabilidad en un sistema, hasta que ésta se corrige.
  - Durante este tiempo, el sistema está sujeto a una amenaza.
- Hay que intentar que la ventana de exposición sea lo más pequeña posible.

Problemas:

- Cuando detectamos una vulnerabilidad, sabemos que la ventana está abierta, pero desconocemos el momento concreto en el que se abrió.
- Alguien puede haber detectado la vulnerabilidad antes que nosotros, y estar explotándola sin hacerla pública.

Ciclo de vida de la ventana de exposición



Reducción de la ventana de exposición

Existen diferentes estrategias:

Publicación inmediata *(full disclosure)*:

- Permite una corrección rápida.
- Suelen aparecer (y corregirse) más fallos.
- Tiene sentido cuando los sistemas son críticos y sus usuarios tienen capacidad para tomar medidas por sí mismos.

Publicación *responsable*:

- El descubridor informa en secreto al desarrollador, y le da un plazo para corregir la vulnerabilidad.
- Pasado el plazo, la vulnerabilidad se publica, usualmente junto al parche. También se publica si se descubre que está siendo explotada.
- Si los plazos son lo suficientemente cortos, es una alternativa válida.

Índice

La seguridad no es un producto

$$\begin{aligned} \text{Sistema de Información} &\longrightarrow \text{entidad dinámica.} \\ \text{Seguridad} &\longrightarrow \text{entidad dinámica.} \end{aligned}$$

- No existe un sistema seguro o inseguro.
  - Lo que es seguro o inseguro es su funcionamiento.
- Un producto siempre es inseguro.
  - Desconocemos sus vulnerabilidades y riesgos cuando se empieza a utilizar.

La seguridad total no existe

- Nuestros entornos de trabajo son intrínsecamente inseguros.
- No hay que confiar en *soluciones milagrosas*.
- En su lugar, hay que...
  - minimizar el número de vulnerabilidades.
  - estar preparados para que el daño sea mínimo en caso de fallo.
    - Controlando los diseños.
    - No asumiendo riesgos innecesarios.
    - Aprendiendo de errores pasados.

Sería interesante que, al igual que ocurre en otros ámbitos, los fabricantes se responsabilizaran de los daños producidos por productos defectuosos.

¿Por qué no se tiene más en cuenta la seguridad?

Se trata de un problema de costes:

- Las restricciones de seguridad hacen más caro y lento el desarrollo de un producto *software*.
- La competencia feroz hace que salgan al mercado productos incompletos, con una calidad discutible.
- Las pruebas *beta* no sirven para detectar vulnerabilidades en un sistema.
  - Normalmente, las vulnerabilidades se detectan a través de pruebas muy complejas y específicas, o de manera fortuita.
- Los diseñadores, desarrolladores y directivos de las empresas carecen de formación sólida sobre seguridad.
- En Informática, está demostrado que un buen *marketing* es más efectivo y rentable que fabricar un producto de calidad.

La seguridad como proceso

- La seguridad no puede entenderse como un producto, sino como un proceso que debe estar presente en todas las fases del ciclo de vida de un sistema:
  - El diseño.
  - El desarrollo e implantación.
  - La definición de cómo debe usarse el sistema.
    - Decidir qué cosas van a estar permitidas y cuáles no.
    - Decidir qué elementos deberán ser supervisados, de qué manera y con qué frecuencia.
  - El uso del sistema propiamente dicho.

La seguridad es un proceso, no un producto

- Todos los sistemas tienen vulnerabilidades y fallos.
- Hay que dar por hecho que se van a producir fallos.
- Es necesaro tener previsto cómo nos vamos a enfrentar a los problemas:
  - Políticas de copias de seguridad, para poder recuperarse frente a pérdidas de datos.
  - Análisis de los registros del sistema para averiguar cómo se produjo un fallo.
  - Detección de eventos *sospechosos* para anticiparse a los fallos.
  - Revisión constante de la organización del sistema.
  - ...

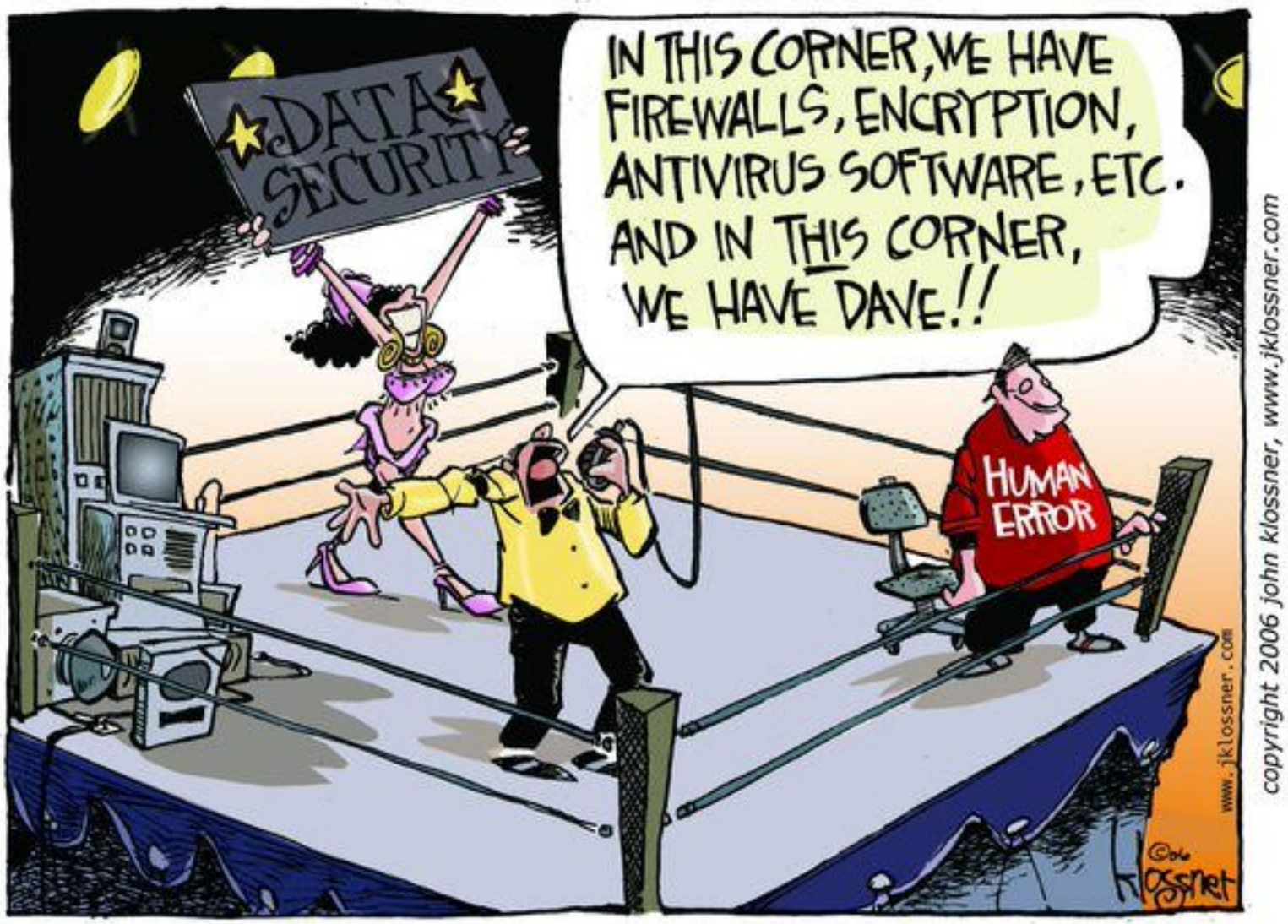

OUR DEVICES ARE NOW 100% SECURE.
I TURNED THEM ALL OFF.
HOW DID YOU DO THAT?
© D.Fletcher for CloudTweaks.com

One Devloper Army
@OneDevloperArmy

Follow

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

11:55 AM - 17 Jul 2018


DATA SECURITY
IN THIS CORNER, WE HAVE FIREWALLS, ENCRYPTION, ANTIVIRUS SOFTWARE, ETC.
AND IN THIS CORNER, WE HAVE DAVE!!
HUMAN ERROR
www.jklossner.com
copyright 2006 john klossner, www.jklossner.com