



Caja de Ahorros  
del Mediterráneo

OBRAS SOCIALES

# Las 5 mentiras de la seguridad web

**Gonzalo Álvarez Marañón**

*Consejo Superior de Investigaciones Científicas*

**Seguridad en Tecnologías de la Información**

# Contenido



- I. Bienvenido al mundo real, Neo
- II. Pero, ¿qué son los ataques web?
- III. Medidas de protección
- IV. Conclusiones

# Cree lo increíble



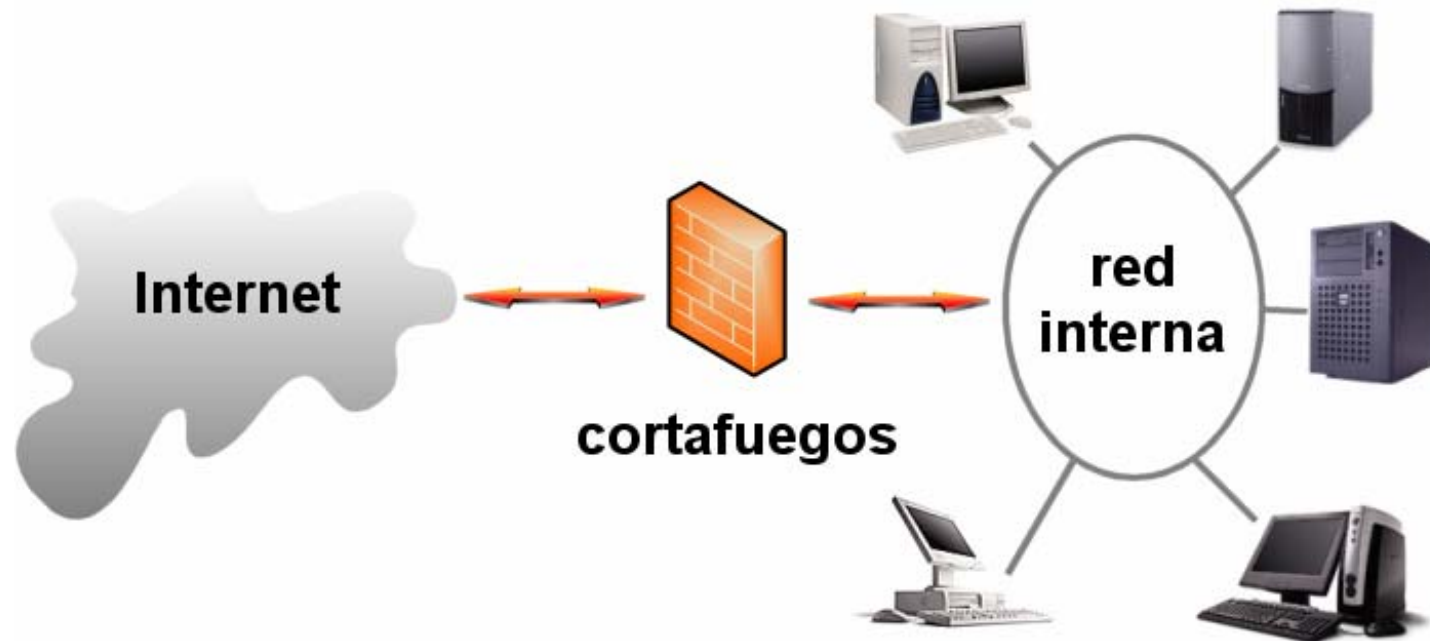
- El mundo real **no** es lo que nos han enseñado a creer
  - Cortafuegos
  - SSL
  - Bastionado/Parches
  - Arquitectura segura de red
  - Auditorías de seguridad



# Cortafuegos



- Aísla la red privada de Internet
- Sólo se permite acceso a unos servicios y el resto se prohíbe



# Cortafuegos

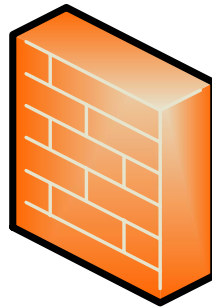


## o Ventajas

- ❑ **Sólo** deja abiertos puerto 80 y 443
- ❑ Detiene algunos ataques **DoS**

## o Limitaciones

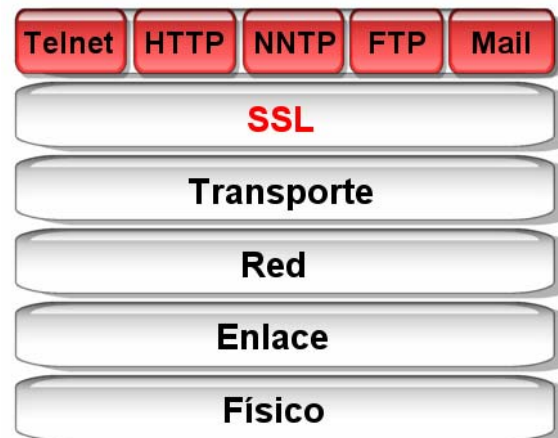
- ❑ No entienden el significado de **HTTP**
- ❑ Dejan pasar el **100%** de los ataques web



# SSL



- Confidencialidad: cifrado de datos
- Integridad de mensajes
- Autenticación de servidores
- Autenticación de cliente (opcional)



# SSL



## o Ventajas

- ❑ Cifra el contenido de las comunicaciones: **canal** seguro
- ❑ Permite autenticar servidores (y clientes)

## o Limitaciones

- ❑ Sólo protege los datos en **tránsito**, no en origen ni destino
- ❑ Los ataques web **pasan** cifrados, pero pasan
- ❑ Dificulta la labor del **IDS**



# Bastionado/Parches



## o Bastionado de la plataforma

- ❑ **SO**: eliminación de servicios innecesarios, configuración de permisos, eliminación de archivos, etc.
- ❑ **Web**: eliminación de extras, borrado de aplicaciones de ejemplo, configuración segura, etc.

## o Aplicación de parches

- ❑ Solucionan vulnerabilidades descubiertas en los productos
- ❑ Actualización religiosa de parches



# Bastionado/Parches



## o Ventajas

- ❑ **Fácil** bastionar: listas de comprobación, herramientas automatizadas, etc.
- ❑ **Fácil** actualizar: herramientas de gestión de actualizaciones
- ❑ Evitan ataques ya conocidos y solucionados

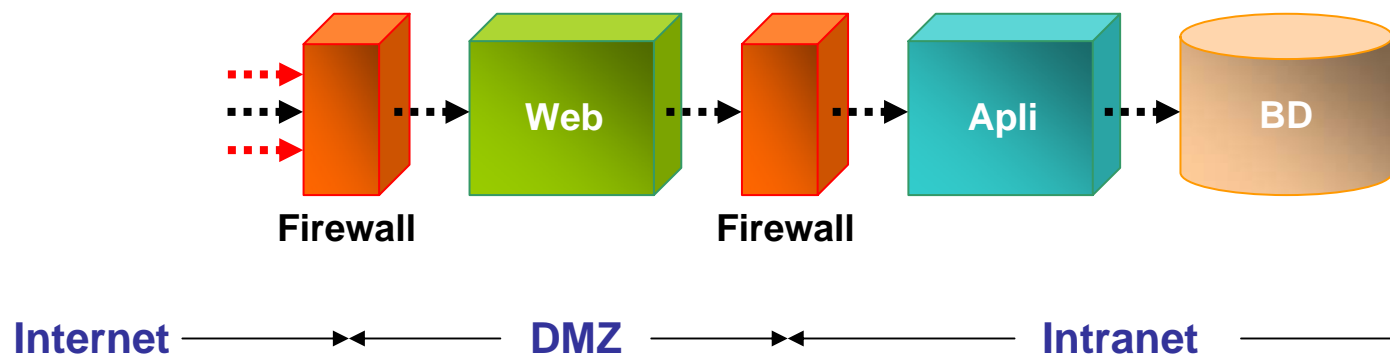
## o Limitaciones

- ❑ Los parches llegan **tarde** y no solucionan el **mal** hecho
- ❑ Las aplicaciones web **no** se parchean
- ❑ La mayoría de ataques web no explotan agujeros de la plataforma, sino de la aplicación

# Arquitectura segura de red



- o **Separación** por funcionalidad → separación de amenazas



# Arquitectura segura de red



## o Ventajas

- Será (casi) imposible acceder **directamente** a la BD

## o Limitaciones

- Existen caminos **indirectos** para acceder a la BD a través de la aplicación
  - Inyección de **SQL**
  - Manipulación de **sesiones**
  - Acceso a los **logs**



# Auditorías de seguridad



## ○ Auditorías automatizadas **periódicas**

- ❑ SO, Red, BD, Web
- ❑ Realizadas con herramientas automatizadas: Nessus
- ❑ Solucionan la **mitad** del problema

## ○ Auditorías manuales externas **puntuales**

- ❑ Caja negra/gris/blanca
- ❑ Completas
- ❑ Solucionan el problema **una** vez



# Auditorías de seguridad



## o Ventajas

- Independencia
- Experiencia en seguridad

## o Limitaciones

- Precio muy elevado
- Los sitios web cambian frecuentemente
- Sólo se realizan una vez

Nuevas vulnerabilidades / Nuevo código

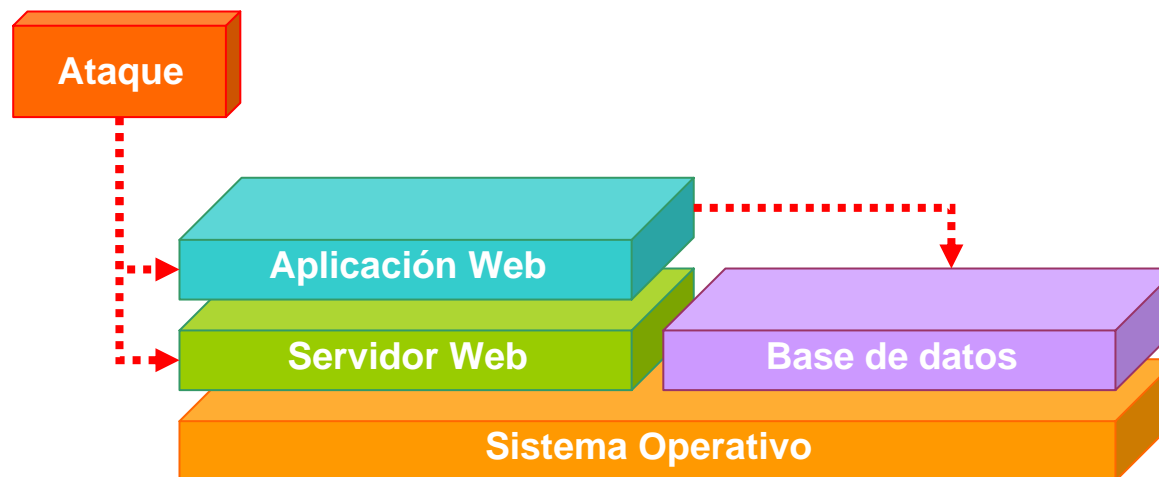
1 semana después

1 año después

# ¿Qué son los ataques web?



- Vulnerabilidades en la aplicación web
- Vulnerabilidades en el servidor web



# OWASP Top 10



1. Entrada de datos sin validar
2. Violación de control de acceso
3. Violación de administración de sesión y autenticación
4. Vulnerabilidades de XSS
5. Desbordamiento de búfer
6. Vulnerabilidades de inyección: comandos, SQL
7. Gestión deficiente de excepciones
8. Almacenamiento inseguro
9. Denegación de servicio
10. Administración de configuración insegura

# Cómo protegerse



- o **Construir** de forma segura
  - Diseño/Codificación seguros
- o **Averiguar** si lo construido es seguro
  - Auditorías de seguridad
- o **Asegurar** lo que ya está construido
  - Cortafuegos de aplicación



# Diseño/Codificación seguros



- Los programadores deberían conocer sus herramientas, lenguajes y plataformas
- La seguridad debería introducirse desde el diseño de la aplicación, no al final
- Los desarrolladores deberían estar concienciados en seguridad
  - El BO es el error más extendido y el más peligroso
  - El XSS es ubicuo y nada inofensivo
  - La inyección de SQL es muy frecuente y peligrosa

# Cortafuegos de aplicación



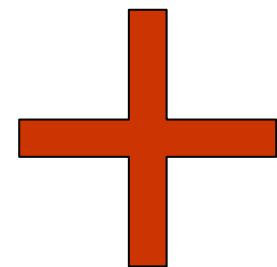
## o Basado en **firmas**

- ❑ Base de datos con patrones de ataques
- ❑ Se buscan coincidencias en la petición
- ❑ Enfoque **negativo**



## o Basado en **anomalías**

- ❑ Definición del comportamiento normal
- ❑ Se buscan peticiones anómalas
- ❑ Inteligencia artificial para aprendizaje
- ❑ Enfoque **positivo**



# Cortafuegos de aplicación

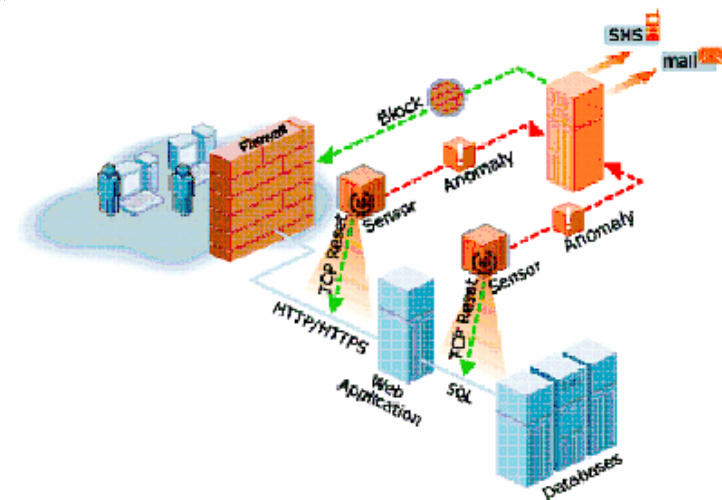


## o Ventajas

- ❑ Detecta los ataques **antes** de que lleguen al servidor en oposición a *análisis de logs*
- ❑ El CISO alcanza **paz** de espíritu

## o Inconvenientes

- ❑ No detecta ataques **lógicos**
- ❑ Problemas de **escalamiento**
- ❑ Tan bueno como su **directiva**



# Conclusiones



- Los controles de seguridad **tradicionales** (cortafuegos, SSL, bastionado, parches, escaneos rutinarios, arquitectura segura) **no** detienen ataques web
- Otras medidas de seguridad (codificación segura, cortafuegos de aplicación) protegen contra ciertas vulnerabilidades, no todas
- No es posible estar 100% seguros, pero sí es posible gestionar el riesgo

<http://www.iec.csic.es/gonzalo>