La seguridad desde el inicio

Introducción

- La seguridad debe introducirse en las etapas iniciales del diseño
- Si bien, esto conlleva recursos y tiempo adicionales
- Posible resistencia:
  - No hay tiempo disponible
  - No aporta valor comercial
  - Requiere personal adicional
  - No hay garantías de ataque
- Estas cuestiones no son inválidas desde un punto de vista comercial
  - No se puede considerar exclusivamente desde el punto de vista de la seguridad
- Esta oposición se resume en la pregunta:

¿Cuál es el valor añadido al ser proactivo en la seguridad del software que desarrollamos?

Mitos de seguridad

1. Tenemos un firewall
2. Usamos SSL
3. Tenemos IDPSs
4. Nuestro software no es accesible desde internet
5. Nunca hemos sido comprometidos
6. La seguridad no es “mi trabajo”, es responsabilidad del proveedor
7. La seguridad aporta escasos beneficios al negocio

#1 Tenemos un firewall

- Es una de las excusas más habituales
- Proviene del modo en que se ha implementado la seguridad de forma histórica.
- Originalmente, había una frontera clara entre el mundo exterior y la compañía.
- El rol de un profesional de la seguridad era la configuración de la seguridad de red.
  [administrador de firewall]

- Caso 1: se elige el personal de seguridad del software de entre el equipo de seguridad en red
  - Todas las soluciones consisten en implementar un firewall
  - El equipo de desarrollo se vuelve "complaciente"
- Caso 2: un conferenciante comenta que "la era del hacker de red está en decadencia"
  - Recibe "hatemail" del público comentando que no tiene ni idea.
  - El público consistía principalmente en administradores de firewall
  - El 70% de los ataques son a la capa de aplicación (2005)

El firewall no es suficiente

Sin fronteras

- En muchos casos, las fronteras son inexistentes
- En cloud computing se consumen plataformas, infraestructuras y software bajo demanda
- En el caso de clouds públicas o híbridas los datos no se almacenan en el perímetro de la compañía
- El firewall de la compañía no ofrece ninguna protección en estos casos

El enemigo interno

- El firewall no protege frente a ataques que se originan desde dentro de la compañía
- Ejemplos:
  - Un empleado descontento
  - Un agente infiltrado por un competidor
  - Un desarrollador introduce “bombas lógicas”

El firewall no es suficiente

- Los sistemas de control perimetral (firewalls, etc.) tienen su lugar como primera línea de defensa

- NO pueden ser el único elemento para proteger las aplicaciones internas

- Son excelentes para:
  - Filtrado de entrada
  - Evitar denegación de servicio

- No sirven para:
  - Filtrado de salida (sistemas DLP)
  - Robo de datos o ataques relacionados con datos

Es necesario mejorar la seguridad de las aplicaciones internas

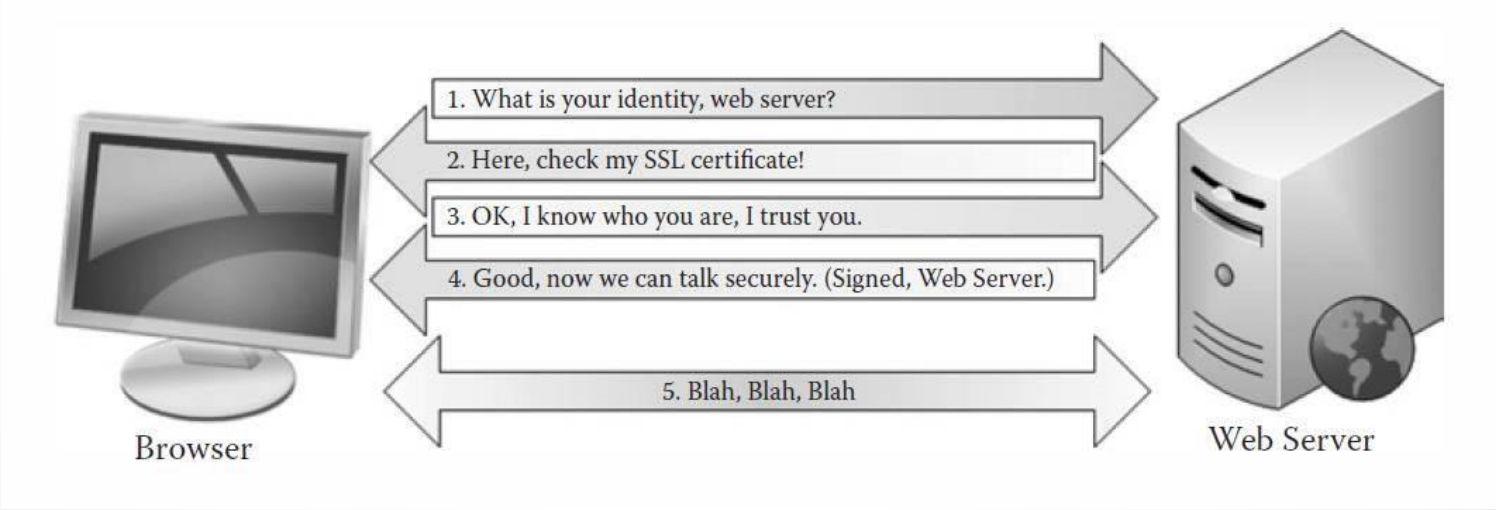
#2 Usamos SSL

- Es, posiblemente, la segunda excusa más habitual:

"Usamos SSL así que debe ser seguro"

- SSL (Secure Sockets Layer) es un protocolo que crea un túnel cifrado entre dos puntos

- El sistema de certificados preinstalado permite evitar el ataque MITM
- No es la panacea de la seguridad
- Evita que el firewall inspeccione el tráfico



SSL no es suficiente

- Caso 1: Smak (Carlos Salgado Jr) es detenido a finales de los 90 por intentar vender 100.000 números de tarjetas de crédito robadas por 260.000$
- Las webs de donde procedían estaban protegidas por SSL
- Smak usó vulnerabilidades del sistema operativo y colocó *sniffers* en puntos clave

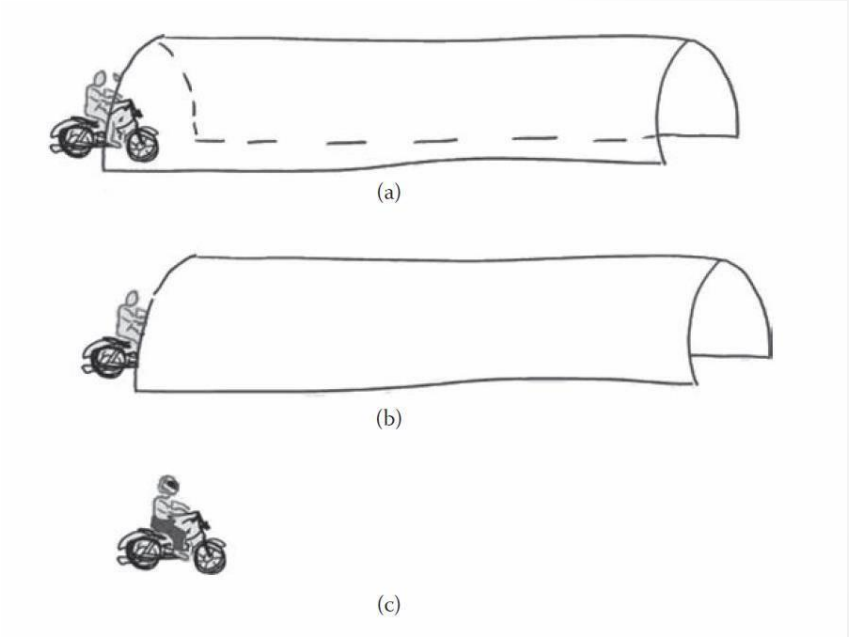
- Caso 2: Moxie Marlinspike, un investigador de seguridad, explica en Black Hat 2009 ataques sobre SSL utilizando MITM, spoofing y HTTP stripping
- No ofrece protección frente a ataques como inyección SQL o Cross-Site Scripting (XSS)
- Hay ataques directos a la criptografía (BEAST 2011, RC4 2013)

SSL no es suficiente

a) El motorista pasa desnudo por un túnel transparente

b) El motorista pasa desnudo por un túnel opaco

c) El motorista va vestido, protegiendo su intimidad incluso en la ausencia de túnel



#3 Tenemos IDPSs

- Estos sistemas sirven de poco si la información que obtienen no se monitoriza y se reacciona en consecuencia
- La mayoría de IDPS proporciona alertas acerca de elementos desde inocuos hasta verdaderamente maliciosos
- La ingente cantidad de información es tal, que peligros reales pueden quedar ocultos en los voluminosos logs
- El análisis manual de dichos logs resulta imposible, haciendo necesaria la automatización.
- La calidad del sistema es directamente proporcional a la configuración de filtros y la preparación del personal que monitoriza los logs
- Los IDPSs son necesarios, pero complementan otras defensas como controles perimetrales, protección frente a malware y endurecimiento de aplicaciones

#4 Software no accesible desde internet

- La máquina más segura es aquella que no esta conectada a internet… y apagada
- Un concepción errónea es que el sistema es seguro puesto que el software interno no es accesible desde internet
- No tiene en cuenta ataques desde el interior

[saltando el gap: Stuxnet]

- Estudios actuales revelan que el número de ataques externos e internos es similar
- No es tan importante el número de ataques como el daño que pueden causar
- Esta forma de pensar provoca una falsa sensación de seguridad
- Mito de los 80: “Mis usuarios son demasiado estúpidos para causar un problema serio”

#5 Nunca hemos sido comprometidos

- Siempre se piensa que sólo le ocurre a otros
- Debemos ser seguros puesto que no nos han atacado todavía
- Microsoft indica como ley #1:
  - Nadie piensa que le va a suceder algo malo hasta que le ocurre
- Cuando ocurre, el daño es devastador

Un gramo de prevención equivale a un kilo de cura

- No hay que caer en la seguridad a través del FUD
  (fear, uncertainty, and doubt)
- Hay que estar preparado para prevenir y mitigar los daños en caso de ataque:
  - Elementos de defensa
  - Preparación del personal
  - Sistemas de backup y logs
  - Plan de actuación
  - Etc.

#6 La seguridad es el trabajo del proveedor

- Al adoptar estrategias de cloud computing (IaaS, PaaS, SaaS,…) se tiende a considerar que la seguridad no es necesaria al ser responsabilidad del proveedor
- Los proveedores de cloud computing tienen expectativas en sentido contrario
- Los proveedores asignan un ~10% de sus recursos a la seguridad
- La actitud de “la seguridad no es cosa mía” produce aplicaciones con escasa seguridad
- En caso de fallo de seguridad, es la empresa quien queda expuesta ante sus clientes y no el proveedor
- Existen múltiples peligros con este tipo de soluciones:
  - NSA, GCHQ, CNI,…
  - Competidores
  - Pérdida de datos
  - Etc.

#7 La seguridad aporta escasos beneficios al negocio

- Adam Smith:
  la paradoja agua-diamante
  oferta / demanda

- Aunque la seguridad es útil para la propia existencia de la empresa, parece que no puede "comprar" tanto como la funcionalidad

- No obstante, para una compañía que ha sido atacada de forma satisfactoria, la seguridad es un valor esencial

- La seguridad no suele ser visible al cliente a modo de funcionalidad. Su valor percibido es menor.

- La realidad es que la seguridad añade valor en términos de permitir a la empresa continuar operaciones y generar el valor de negocio esperado

- Si la web es atacada y "degradada", la empresa no puede continuar su actividad comercial y sufre pérdidas. La seguridad evita que esto ocurra

#7 La seguridad aporta escasos beneficios al negocio

- El valor de la seguridad debe tomarse desde una perspectiva de ahorro además de una basada en el retorno a la inversión (ROI)

- Si bien el ROI en seguridad es difícil de observar de forma directa, el valor obtenido en ahorro de costes es significativo

- Ahorro en múltiples vías:
  - Coste de reparación de vulnerabilidades
  - Coste de reparación de reputación o imagen de marca

Seguridad desde el inicio: necesidad

- Las listas de seguridad y vulnerabilidades son un claro ejemplo de que nuestras aplicaciones no son seguras
- La lista de empresas que han sido víctimas de ataques de seguridad incluye grandes compañías no sólo peces pequeños y en múltiples sectores
- El problema no surge exclusivamente de la metodología de desarrollo de software, también de las personas y las tecnologías empleadas

[enfoque holístico]

- El coste anual de 180M$ relacionado con el software inseguro es significativo
- Ataque a Sony: 100M de cuentas de usuario robadas, coste estimado en billones de dólares. El coste de imagen es todavía mayor
- [Ataque a Adobe]

Seguridad desde el inicio: necesidad

- La motivación de los ataques ha cambiado:
  - Ataques por “ego”
  - Ataques por dinero
- Existe una economía de servicios basada en el hacking criminal
  - Los beneficios superan los riesgos (con creces)
- También existe el hacking por motivación política o “causa”
  - Anonymous, Lulzsec
  - Syrian Liberation Army
  - Hacktivismo
- Los ataques se orientan a aplicaciones:
  - Lulzec, inyección SQL contra Sony
  - RefRef (Anonymous), utiliza JavaScript y vulnerabilidades SQL para crear DoS
- Es necesario un nuevo tipo de defensor:
  - Entiende como funciona el software
  - Entiende cómo se rompe el software

Seguridad desde el inicio: elementos

- Hay 3 elementos para perpetrar un crimen:
  - motivación, oportunidad y medios
- Los controles de seguridad pueden reducir la superficie de ataque (oportunidad) y los medios disponibles para comprometer el software
- Poco se puede hacer con la motivación

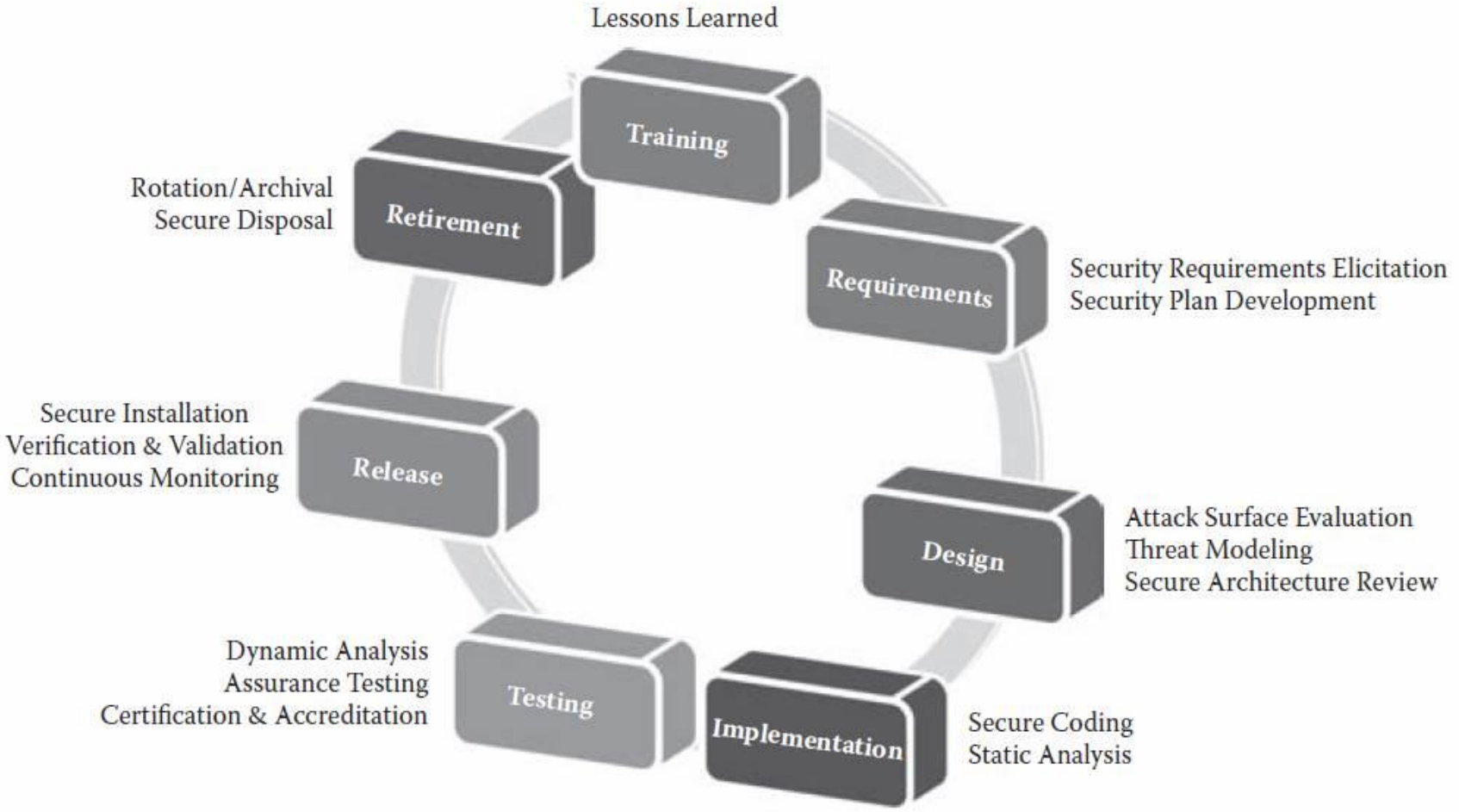
- Hay dos tipos de controles de seguridad:
  - Proactivos
  - Reactivos
- La seguridad desde el inicio implica ser proactivo diseñando y desarrollando los elementos de seguridad necesarios en el software
- Esto se puede lograr a través del personal, el proceso y los componentes tecnológicos del proceso de ingeniería del software

Seguridad desde el inicio: elementos

- Actualmente, la seguridad de las aplicaciones se ve como una prioridad; si bien, hay poca inversión
- Sería necesario un cambio en el comportamiento poniendo importancia en la gestión proactiva del riesgo y no exclusivamente en el ROI
- Dicho cambio requiere planificación estratégica y compromiso de largo plazo

- Desafortunadamente, la mayoría de los programas de seguridad actuales se basan exclusivamente en herramientas ad-hoc *[enfoque táctico]*
- La seguridad desde el inicio implica:
  - Establecer requisitos de seguridad
  - Modelar riesgos de seguridad para definir la superficie de ataque
  - Revisar código para encontrar vulnerabilidades
  - Validar la efectividad de los controles de seguridad
  - Configurar e instalar la aplicación detectando puntos de entrada

Seguridad desde el inicio: SDLC



Seguridad desde el inicio: valor añadido

- Al introducir la seguridad en el diseño del software obtenemos:
  - Fiabilidad
    (funciona como se espera)
  - Resiliencia
    (resiste abuso y ataques)
  - Recuperabilidad
    (restauración de actividad comercial)

- Se reduce el coste asociado a la reparación de defectos
- Se minimiza la posibilidad de que un atacante puede comprometer el software o los datos que procesa
- Es imposible alcanzar software seguro al 100%, pero es necesario dificultar el proceso

[camino de menor resistencia]

Seguridad desde el inicio: resumen

- Es fiable
- Es resiliente
- Es recuperable
- Menor posibilidad de error
- Menor posibilidad de publicar información sensible
- Está disponible cuando se necesita
- Se diseña bajo especificaciones funcionales y de seguridad
- Es menos susceptible a fallos lógicos o semánticos
- Cumple las regulaciones establecidas
- Se ha modelado el riesgo y se conoce la superficie de ataque
- Su superficie de ataque relativa es reducida
- Es seguro frente a ataques comunes
- Ha sido auditado en busca de vulnerabilidades
- Se ha instalado y configurado de forma adecuada
- Se monitoriza y actualiza regularmente
- Trata y elimina los datos de forma segura