DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Fundamentos de ciberseguridad

Seguridad en el Ciclo de vida del desarrollo de software (SDLC)

Profesor

Mg. Juan Ignacio Iturbe A.

Ciclo de vida del software (SDLC)

- El ciclo de vida del software lidia con establecer un proceso repetible y predecible en el tiempo que asegure:
  - Funcionalidad
  - Costo
  - Calidad
  - Tiempo de entrega respetados.
- ¿Cómo nos podemos asegurar que se construye el mejor producto de software posible?

Ciclo de vida del software (SDLC)

- Existen muchos acercamientos de modelos SDLC pero la mayoría de ellos, consideran los siguientes aspectos:
  - Obtención de requerimientos
  - Diseño
  - Desarrollo
  - Pruebas/Validación
  - Liberación/Mantenimiento

Gestión de proyectos

- La buena gestión de proyecto mantiene el proyecto en movimiento en el dirección correcta.
- El proceso de gestión de proyectos debe estar en su lugar para que un proyecto de desarrollo de software ejecute cada fase del ciclo de vida.
- La gestión de la seguridad es una importante parte de la gestión de proyectos.

Gestión de proyectos

- Un plan de seguridad
  - debe elaborarse a principios del proyecto de desarrollo e integrarse en el plan funcional para garantizar que la seguridad no se pasa por alto
  - Tiene su propio tiempo de vida. Estos tiempos deben ser considerados en el proyecto.
  - Debe ser revisado para que las decisiones asociadas puedan ser entendidas.

Fase de obtención de requerimientos

- Se intenta entender porque el proyecto es necesario y cual es el alcance del mismo.
- El equipo examina los requerimientos de software y proponen las funcionalidades.
- Esta fase puede incluir la evaluación de productos en el mercado e identificación de demandas no tomadas en cuenta por los vendedores.

Fase de obtención de requerimientos

- En lo concerniente a la seguridad, los siguientes puntos deben ser contemplados:
  - Requerimientos de seguridad
    - ¿Qué tipo de seguridad es requerida para el producto de software y en que grado?
  - Evaluación de riesgos de seguridad
    - Debe ser realizado para para identificar amenazas potenciales y sus consecuencias asociadas.
    - Este proceso generalmente implica hacer muchas preguntas para elaborar la lista de vulnerabilidades y amenazas

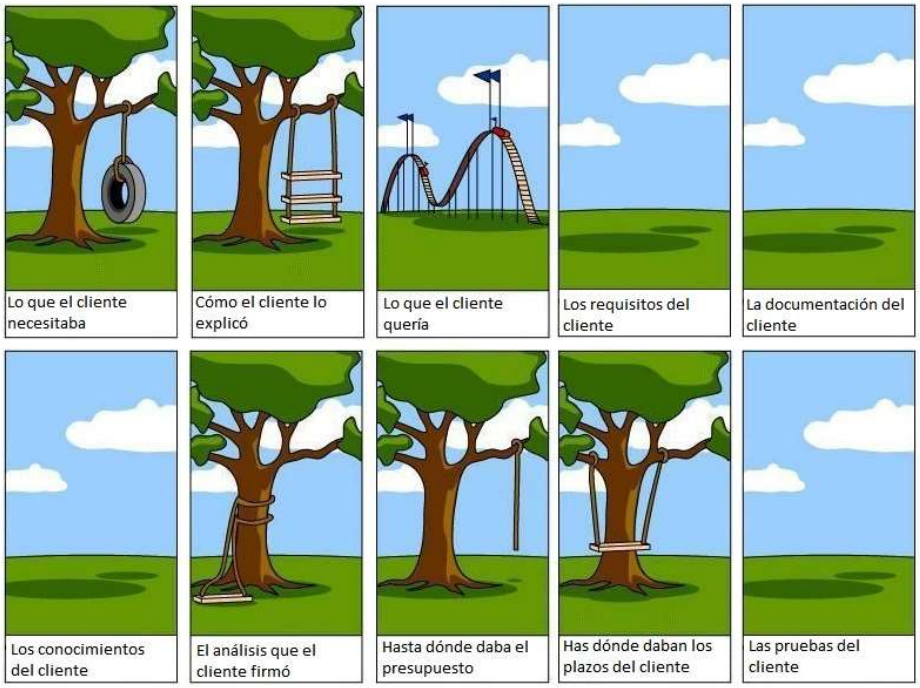
Fase de obtención de requerimientos

- Evaluación de riesgos de privacidad
  - El nivel de sensibilidad de los datos del software debe ser mantenida y procesado. Esto ha ido incrementando en importancia a través de los años.
  - Después de la evaluación de riesgos de privacidad, se debe asignar un *Rating de impacto de privacidad*, que debe indicar el nivel de sensibilidad de la data a procesar o acceder. Por ejm:
    - P1 Riesgo de privacidad alto: se almacena o transfiere Información de identificación personal (PII).
    - P2 Riesgo de privacidad moderado: existe un comportamiento iniciado por el usuario que transfiere datos anónimos.
    - P3 Riesgo de privacidad bajo: No se afecta a la privacidad. No hay datos anónimos o datos personales que se transfieran o almacenen en la maquina.

Fase de obtención de requerimientos

- Aceptación del nivel de riesgo
  - Un criterio claro de aceptación del riesgo debe ser establecido. Por ejemplo:
    - Se aceptaran los niveles de riesgos muy bajo, bajo y medio.
    - Se aceptaran los riesgos talque sus controles mitigantes, excedan la capacidad económica de la empresa (análisis costo-beneficio)

Fase de diseño



- Esta fase comienza a mapear la teoría con la realidad.
- La teoría abarca todos los requerimientos identificados previamente
- El diseño delinea como el producto va a cumplir estos requerimientos.

Fase de diseño

- Los requerimientos de software comúnmente vienen desde tres modelos:
  - Modelo informativo: dicta el tipo de información a ser procesada y como debe ser procesada
  - Modelo funcional: delinea las tareas y funciones que la aplicación necesita llevar a cabo
  - Modelo de comportamiento: explica los estado que la aplicación debe tener durante y despues de transiciones especificas se lleven a cabo.

Fase de diseño

- Por ejemplo un antivirus:
  - El modelo de informativo dicta que la información va a ser procesada por el programa, como firmas de virus, archivos de sistemas modificados, checksums en archivos críticos y actividad de virus.
  - El modelo funcional dicta que la aplicación debe ser capaz de escanear el HDD, revisar el email para firmas de virus conocidas, monitorear archivos de sistema críticos y actualizarse a sí mismo.
  - El modelo de comportamiento indica que cuando el sistema parte, el antivirus debe escanear el HDD y los segmentos de memoria. Si un virus es encontrado la aplicación debe cambiar de estado y lidiar con el virus apropiadamente.

Fase de diseño

- Desde el punto de vista de la seguridad, los siguientes puntos deben ser cumplidos:
  - Análisis de superficie de ataque
  - Modelamiento de amenazas

Fase de diseño

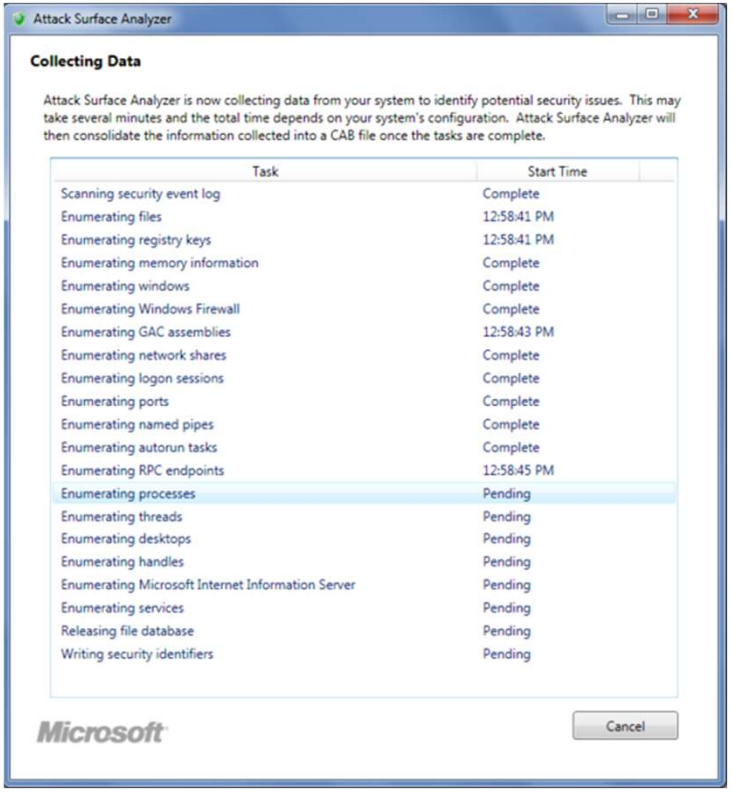
- La superficie de ataque es que esta disponible para ser usado por un atacante en contra del mismo producto.
  - Por ej. Si uno tiene una armadura de mitad de cuerpo, la otra mitad es la mitad con la superficie de ataque vulnerable.
- El equipo desarrollador debe disminuir la superficie de ataque lo mas que sea posible,
  - Entre mayor la superficie de ataque, mas entradas tiene un atacante y por lo tanto mayor probabilidad de éxito.

Fase de diseño

- El objetivo del análisis de la superficie de ataque es identificar y reducir el conjunto de código y funcionalidad accesible por usuarios no confiables.
- La estrategia básica para reducir la superficie de ataque son:
  - Reducir el conjunto de código corriendo.
  - Reducir los puntos de entrada disponibles para usuarios no confiable.
  - Reducir los niveles de privilegio tanto como sea posible
  - Eliminar servicios innecesarios.

Fase de diseño

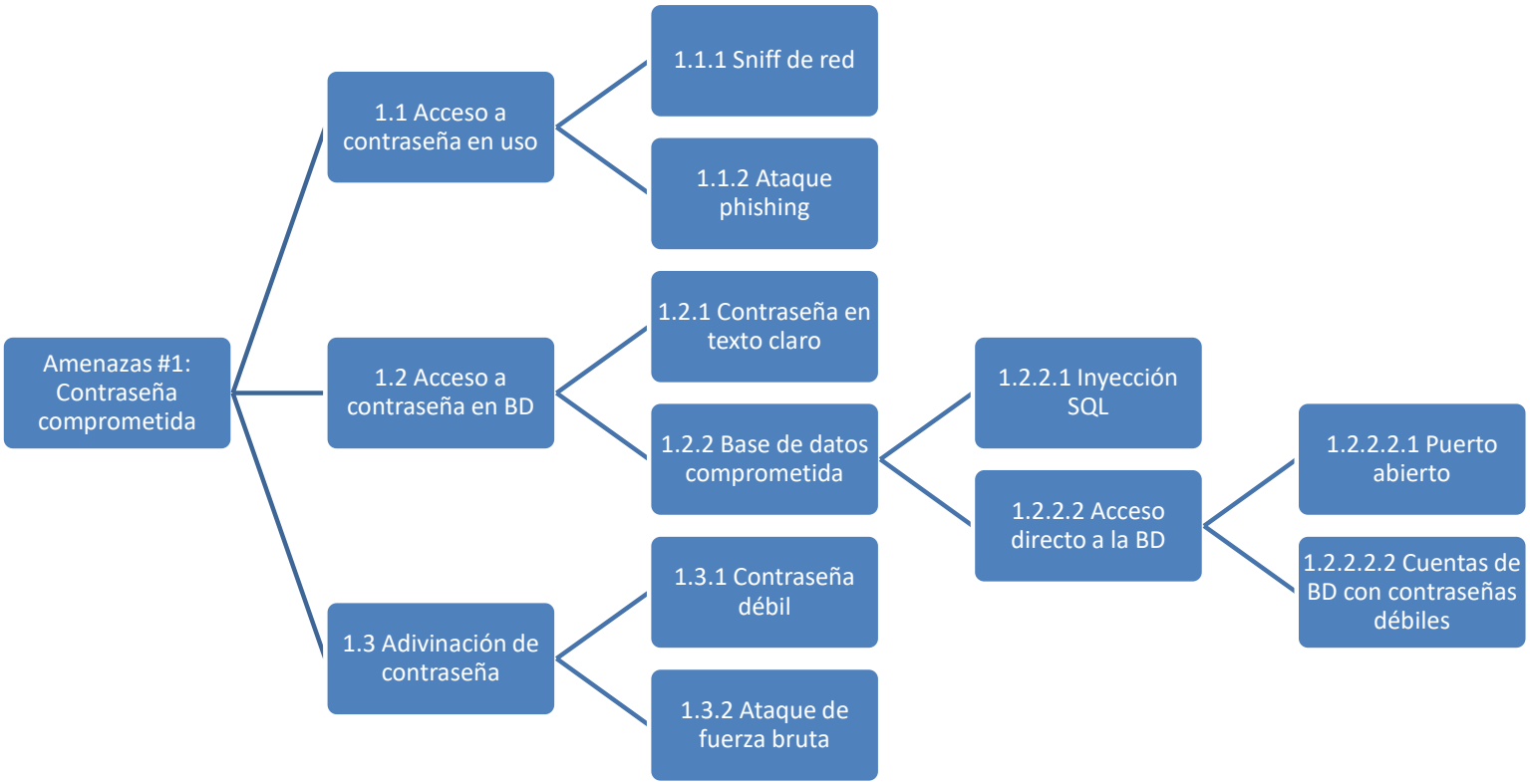
- El análisis de superficie de ataque puede ser realizado por herramientas especializadas. Estas:
  - Revisan archivo
  - Llaves de registros
  - Datos de memoria
  - Información de sesión, procesos
  - Detalles de servicio



Fase de diseño

- Modelamiento de amenazas
  - Se utiliza para entender como las diferentes amenazas pueden ser encontradas y como se puede dar el compromiso del sistema.
  - Se plantean diferentes escenarios de amenazas y se revisan como estas se podrían materializar.
  - Es común en los equipos de desarrollo el uso de arboles de amenazas.
  - Existen varias herramientas automatizadas que se pueden utilizar en este punto.

Fase de diseño



Fase de desarrollo

- Herramientas CASE
- Análisis estático
- Revisión de código
- Sitios con errores comunes
  - Mitre
  - OWASP
  - Etc.

Fase de pruebas y validación

- Análisis dinámico
- Fuzzing
- Pruebas manuales
- Pruebas
  - Unitarias
  - Integración
  - Aceptación
  - Pruebas de regresión

Fase de lanzamiento y mantención

- El software se encuentra listo para ser implementado en el ambiente de producción.
- Nuevos problemas y vulnerabilidades se encontrarán en este punto.
- Pueden aparecer problemas en la interoperabilidad con otros sistemas, el usuario puede detectar algunas vulnerabilidades, etc.
- En este punto aparecen las vulnerabilidades de día cero.

FIN

DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE