

Seguridad en la Red Informática Mundial

Metodologías de desarrollo de
aplicaciones Web seguras

Semana 2 clases 3 y 4

Mtra. María Noemí Araiza Ramírez



¿Qué es un Método?

Es una forma ordenada y sistemática que nos permitirá conseguir una meta o lograr un objetivo.

Procedimiento para alcanzar algo

Por ejemplo:

Definir los pasos necesarios para la automatización de un sistema



¿Qué es una Metodología?

Conjunto de métodos empleados por una disciplina

Una metodología le da énfasis al entorno en el cuál se analiza y estructura el desarrollo de un sistema o una aplicación.

Hay distintas metodologías, aunque no todas son compatibles con nuestros desarrollos, debido a que el ciclo de vida del software puede variar.

Por esta razón, es importante que dependiendo del tipo de la aplicación que se vaya a desarrollar, se identifique la metodología idónea

Metodología de Desarrollo

- Es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de SI.
- Es un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar nuevo software.
- Es una metodología para desarrollar software de manera sistemática.


- Por lo general las metodologías se encuentran documentadas
- Una metodología no aplica necesariamente para cualquier tipo de proyecto, por lo que al conocer cada una se debe pensar en cuál se adaptará mejor
- En inglés se le conoce como Software Development Methodology (SDM)

Objetivos de una Metodología de Desarrollo

- Conseguir el desarrollo de mejores aplicaciones
- Contar con un Proceso de Desarrollo que identifique salidas (o productos intermedios) de cada fase; de forma que se pueda planificar y controlar un proyecto
- Un Proceso Estándar en la organización o de forma particular (desarrolladores independientes)

¿Seguridad en el desarrollo de aplicaciones?

- No es un producto, es un proceso [Bruce Schneier]
- La seguridad en el desarrollo es tan importante como cuando ya se tiene lista la aplicación y se ejecuta por un usuario.
- Para desarrollarla se requiere de un sistema operativo específico, el cual al igual que todos los componentes del ambiente de cómputo, está en riesgo.

- Están disponibles sistemas de seguridad que se utilizan para proteger a los sistemas operativos sobre los cuales se realizan pruebas durante el desarrollo de software
 - La seguridad en una aplicación generalmente evita que se produzcan errores en los sistemas operativos en donde se realiza el desarrollo, así como en las pruebas del funcionamiento de la aplicación
 - Cuando se hacen pruebas, es común que se produzcan fallas permanentes, las cuales afecten al sistema; pero realmente es necesario que se ejecuten para corregirlas, ya que de otro modo no habría forma de detectarlas
- 

¿Qué metodologías de desarrollo de software seguro son las más conocidas en la actualidad?


- Microsoft (Iniciativa Trustworthy Computing)
- OSSTMM (Open Source Security Testing Methodology Manual)
- OWASP (Open Web Application Security Project)
- OASIS Web Application Security (WAS) project

Microsoft (Iniciativa Trustworthy Computing)

Diseñada por Microsoft en el año 2001 como una necesidad del mercado para contar con soluciones que consideren los aspectos de seguridad y aspectos de operación y funcionalidad, con un enfoque de trabajo colaborativo a largo plazo y tiene como objetivo crear y disponer para la comunidad en general una computación basada en mejores prácticas (best practices) que sea más segura, privada y confiable.

Microsoft (Iniciativa Trustworthy Computing)

Este modelo también abarca conceptos como la estabilidad, la confianza y la seguridad en la plataforma y este último concepto se sustenta de 5 pilares primordiales que no debemos dejar de tomar en cuenta.

- Aislamiento y flexibilidad
 - Calidad
 - Autenticación
 - Autorización y control de accesos
 - Orientación y formación
- 

Microsoft (Iniciativa Trustworthy Computing)

El modelo TwC se basa en 4 pilares: Seguridad, Privacidad, Confiabilidad y Mejores prácticas.

Seguridad Se busca garantizar la seguridad de los sistemas y la información que manejan, contra ataques (considerando a la tecnología, las personas y los procesos como base de la Seguridad).

Esto es:

- Son resistentes a ataques
- Protegen la confidencialidad, integridad y disponibilidad de los datos y sistemas.
- Ningún virus atentará contra nuestros sistemas o los volverá inutilizables.

Microsoft (Iniciativa Trustworthy Computing)

Privacidad Se requiere mantener segura la información personal, financiera e identidad de los usuario mientras su control y uso quedan a disposición de su dueño.

Es decir, Los individuos controlan sus datos personales, la información personal no será expuesta de ninguna forma, ni utilizada de manera que no sea la explícitamente indicada, los productos y servicios en línea se adhieren a principios de información básicos.

Microsoft (Iniciativa Trustworthy Computing)

Confiabilidad Se busca desarrollar software confiable para atender las necesidades tanto de usuarios y negocios.

Esto es que cuando instalemos un programa, no provocará efectos colaterales sobre otro software instalado ni sobre el hardware, además cuentan con excelencia en Ingeniería, son confiables y tienen un desempeño acorde a las expectativas y están disponible cuando se necesita.

Microsoft (Iniciativa Trustworthy Computing)

Mejores prácticas Es el objetivo que tiene Microsoft de ser responsable y transparente, trabajando en la excelencia de sus procesos y decisiones, considerando las mejores prácticas que ha probado la industria. La visión de Microsoft es mejorar en estas áreas para generar confianza.

También, tomar en cuenta que se debe tener una interacción abierta y transparente con clientes, solucionar problemas con productos y servicios, les ayuda a los clientes a que encuentren las soluciones adecuadas, el proveedor de servicios responde rápida y efectivamente a los clientes cuando le informen de un problema.

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Surge el **Modelo de Amenazas (Threat Model)** en él vemos que la seguridad debe abarcar todas las fases. Por lo tanto, el software debe ser:

Seguro por Defecto

La aplicación recién instalada tiene un comportamiento suficientemente seguro.

Un ejemplo, es el gestor de áreas de exposición de SQL Server

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Seguro por Diseño

Ninguna parte de la aplicación queda fuera del control de seguridad

Seguro en la Distribución

Informar al usuario sobre la seguridad de la aplicación, mecanismos de modificación de las características de seguridad, la creación de parches de seguridad tan pronto como se detecte una nueva vulnerabilidad.

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Seguro en las Comunicaciones

Se debe tener asegurada la transferencia de los datos y los medios de transmisión para que no se pueda afectar el paso de datos por una red.

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Ventaja del atacante y dilema del defensor

Howard y Leblanc ejemplifican perfectamente el problema en estos cuatro principios:

- El que defiende, debe defender todos los puntos; **el atacante puede seleccionar el más débil**
- El defensor sólo puede defenderse de ataques conocidos; **el atacante puede probar nuevas formas de ataque.**
- El defensor debe de estar en constante estado de vigilancia; **el atacante puede golpear a voluntad.**
- El defensor debe jugar según las reglas; **el atacante puede jugar sucio.**

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

La metodología Microsoft establece cinco fases

1

Identificar activos de la aplicación

2

Crear información general sobre la arquitectura

3

Descomponer la aplicación

4

Identificar, documentar y clasificar las amenazas

5

Identificar las vulnerabilidades

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Debemos considerar las amenazas que nos muestra el modelo STRIDE

Spoofing (Suplantación)

Tampering (Manipulación)

Repudiation (Repudio)

Information Disclosure (Divulgación de Información)

Denial of Service (Denegación de Servicio)

Elevation of privileges (Elevación de privilegios)

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Spoofing (Suplantación)

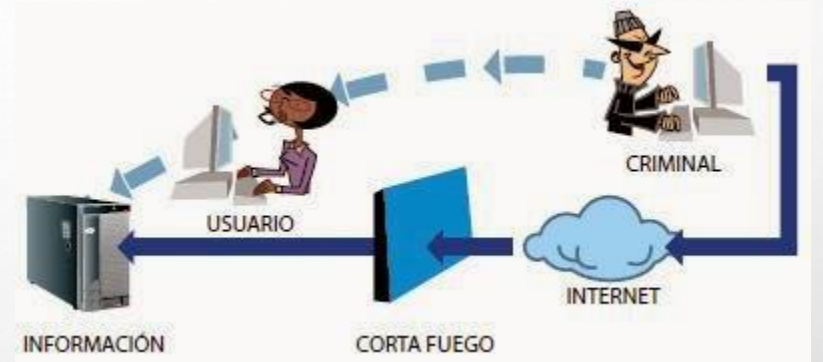
Se adopta la personalidad de otro usuario en un equipo.
Ejemplos:

- Acceso no autorizado a la información de autenticación de otro usuario y su uso, como nombre de usuario y contraseña.
- Enviar mensajes de correo electrónico con una dirección falsa o de otro usuario.
- Crear una cuenta de una red social con información de otro usuario.



Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Tampering (Manipulación)



Es la modificación malintencionada de los datos por parte de algún usuario.
Ejemplos:

- Alterar datos durante su transferencia en una red.
- Cambiar datos en archivos almacenados en un equipo.
- Cambiar datos en una base de datos.

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Repudiation (Repudio)

Son acciones en las que los usuarios niegan su autoría sin que se pueda probar lo contrario. El no rechazo se refiere a la posibilidad de un sistema de contrarrestar las amenazas de repudio.

Ejemplos:

- Un usuario que realiza una operación ilegal sin que existan bitácoras o logs.
- Cuando a un usuario le llega un paquete por mensajería debe firmar un recibo contra entrega, la cual es la prueba de entrega del paquete que tiene el proveedor.



Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Information Disclosure (Divulgación de Información)

Las amenazas de divulgación de información revelan información a personas no autorizadas.

Ejemplos:

- Divulgar información en mensajes de error.
- Lectura de un archivo al que no se debería tener acceso.
- Exponer el código de un sitio web.
- Intruso leyendo datos que están en transferencia entre dos equipos.



Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Denial of Service (Denegación de servicio)

**Denial-of-service
Attack**



Los ataques por denegación de servicio (DoS) ocasionan la pérdida de servicio a los usuarios válidos, por ejemplo, deshabilitando temporalmente un servidor Web, que ponen en riesgo la disponibilidad y fiabilidad del sistema.

Ejemplos:

- Inundar la red con paquetes ICMP falsificados.
- Inundar la red con paquetes de sincronización.

Microsoft (Iniciativa Trustworthy Computing) *Modelo de Amenazas (Threat Model)*

Elevation of privileges (Elevación de privilegios)



En este tipo de amenaza, un usuario sin privilegios obtiene acceso con privilegios, por lo que se pone en peligro a todo el sistema, tanto en su acceso como en la confiabilidad de sus datos.

Ejemplos:

- Explotar la saturación de un búfer para obtener privilegios en el sistema.
- Un usuario que burla las defensas de un sistema y obtiene privilegios de administrador de forma no autorizada.

Directiva DREAD

Cálculo del riesgo DREAD

Fórmula de riesgo DREAD

- Daño Potencial
- Facilidad de **R**eproducción
- Capacidad de **E**xplotación
- Usuarios **A**fectados
- Dificultad para su **D**escubrimiento

- DREAD se usa para formar parte del razonamiento detrás de la clasificación de riesgos, y sirve directamente para ordenar riesgos.
- Cada Riesgo se calcula como un promedio de los cinco
- parámetros anteriores, **valorados de 0 a 10**
- **A mayor número, mayor riesgo**

- **Riesgo DREAD = $(D + R + E + A + D) / 5$**

Directiva DREAD

Daño Potencial
¿Cuánto daño causa?

0 - Nada

5 - La información individual del usuario se ve comprometida.

10 - Destrucción completa del sistema.

Facilidad de Reproducción
¿La amenaza se puede reproducir con facilidad?

0 - Muy difícil o imposible, incluso para los Administradores del sistema

5 - En uno o dos pasos. Quizá requiera de usuario autorizado

10 - Basta con una barra de direcciones y sin estar registrado en el sistema

Capacidad de Explotación
¿Qué se necesita para explotar la amenaza?

0 - Habilidades avanzadas de programación y redes, herramientas de ataque avanzadas o personalizadas

5 - Malware existente, o fácilmente realizado utilizando herramientas normales de ataque

10 - Solamente un navegador

Directiva DREAD

Usuarios **A**fectados
¿Cuántos usuarios se verán afectados por esta amenaza?

0 - Ninguno

5 - Algunos usuarios, pero no todos.

10 - Todos los usuarios.

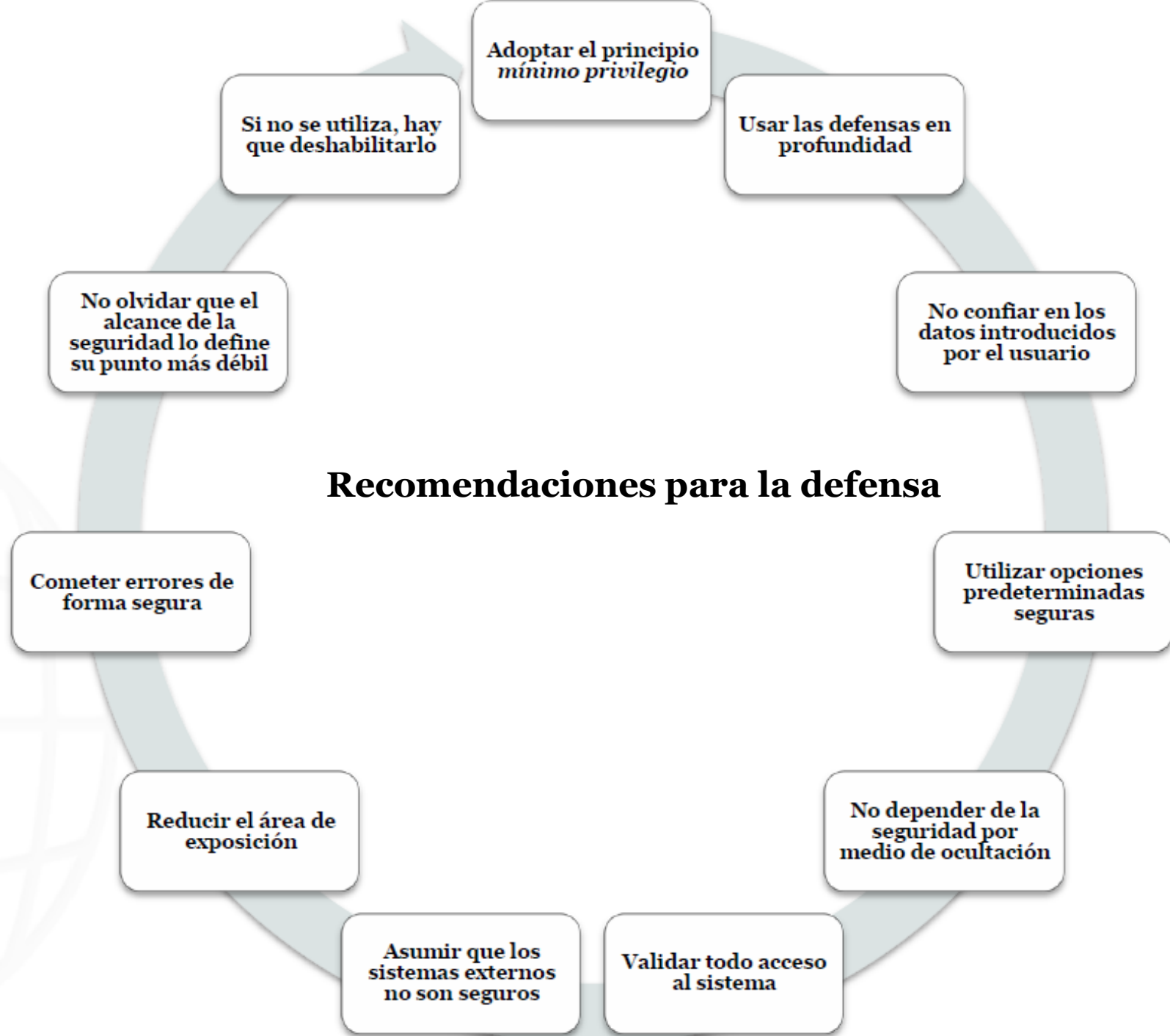
• Dificultad para su **D**escubrimiento
¿Es fácil descubrir la amenaza?

0 - De muy difícil a imposible. Requiere acceso al sistema o al código

5 - Se podría, mediante pruebas u observando la huellas en la red

9 - Los detalles de errores de este tipo son de dominio público y se pueden descubrir con facilidad con cualquier buscador

10 - La información es visible en el la barra de direcciones del navegador Web o en un formulario



Álvarez, Marañón, Gonzalo, and García, Pedro Pablo Pérez. Seguridad informática para empresas y particulares, McGraw-Hill España, 2004. ProQuest Ebook Central,
<http://ebookcentral.proquest.com/lib/univunirsp/detail.action?docID=3195263>.

Escrivá, Gascó, Gema, et al. Seguridad informática, Macmillan Iberia, S.A., 2013. ProQuest Ebook Central,
<http://ebookcentral.proquest.com/lib/univunirsp/detail.action?docID=3217398>.

Unir. (28 de octubre de 2016). Metodologías de desarrollo web seguro. Recuperado el 15 de mayo de 2018