La seguridad en software: ¿un oxímoron?

José María Campaña V.
jose.maria.campana@unam.mx

Contenido

1. Introducción a la seguridad de software
    - Aplicaciones de Web

2. Vulnerabilidades, amenazas y ataques al software

3. Modelos de referencia de seguridad en software

4. Seguridad y SDLC (Software Development Life Cycle)

Introducción a la seguridad de software

La seguridad en software:
¿un oxímoron?

Introducción a la seguridad de software

Contexto de seguridad

Internacional:

- Septiembre de 2001
- Escándalo WikiLeaks

Nacional:

Ley Federal de Protección de Datos Personales en Posesión de Particulares LFPDPPP (julio de 2010):

- Exige controles de seguridad en el ciclo de vida de la información
  - Generación o captación
  - Tratamiento
  - Almacenamiento
  - Transmisión
  - Destrucción (disposición)
- Define los derechos ARCO:
  - Acceso
  - Rectificación
  - Cancelación
  - Oposición

Introducción a la seguridad de software

Estado de la tecnología:

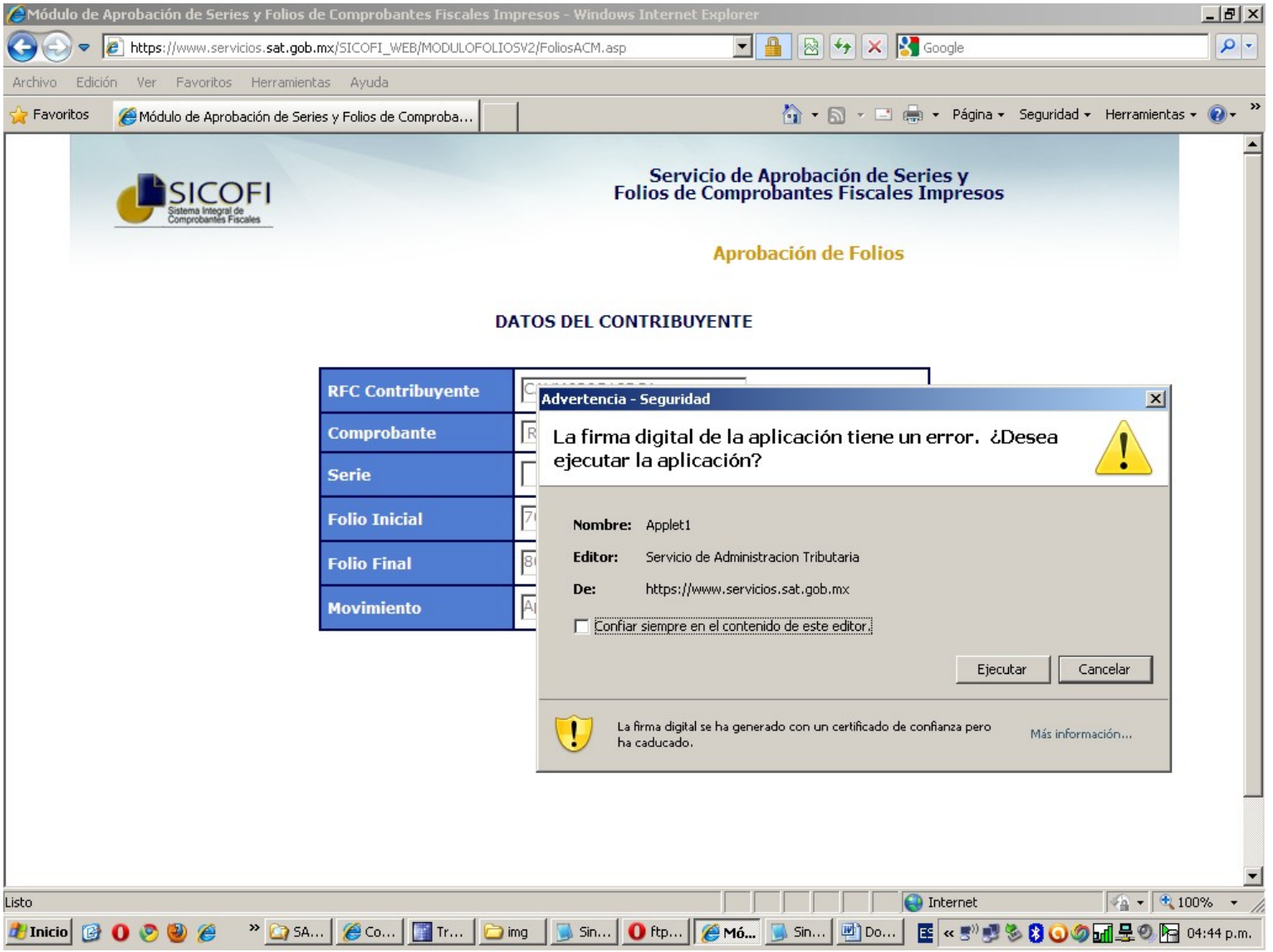
- Software de infraestructura inseguro (navegadores, intérpretes, etc.)

- Reducción drástica en los costos por unidad de almacenamiento
  - Consecuencia directa → Persistencia de los datos
    (¡Todos los datos pueden ser persistentes!)

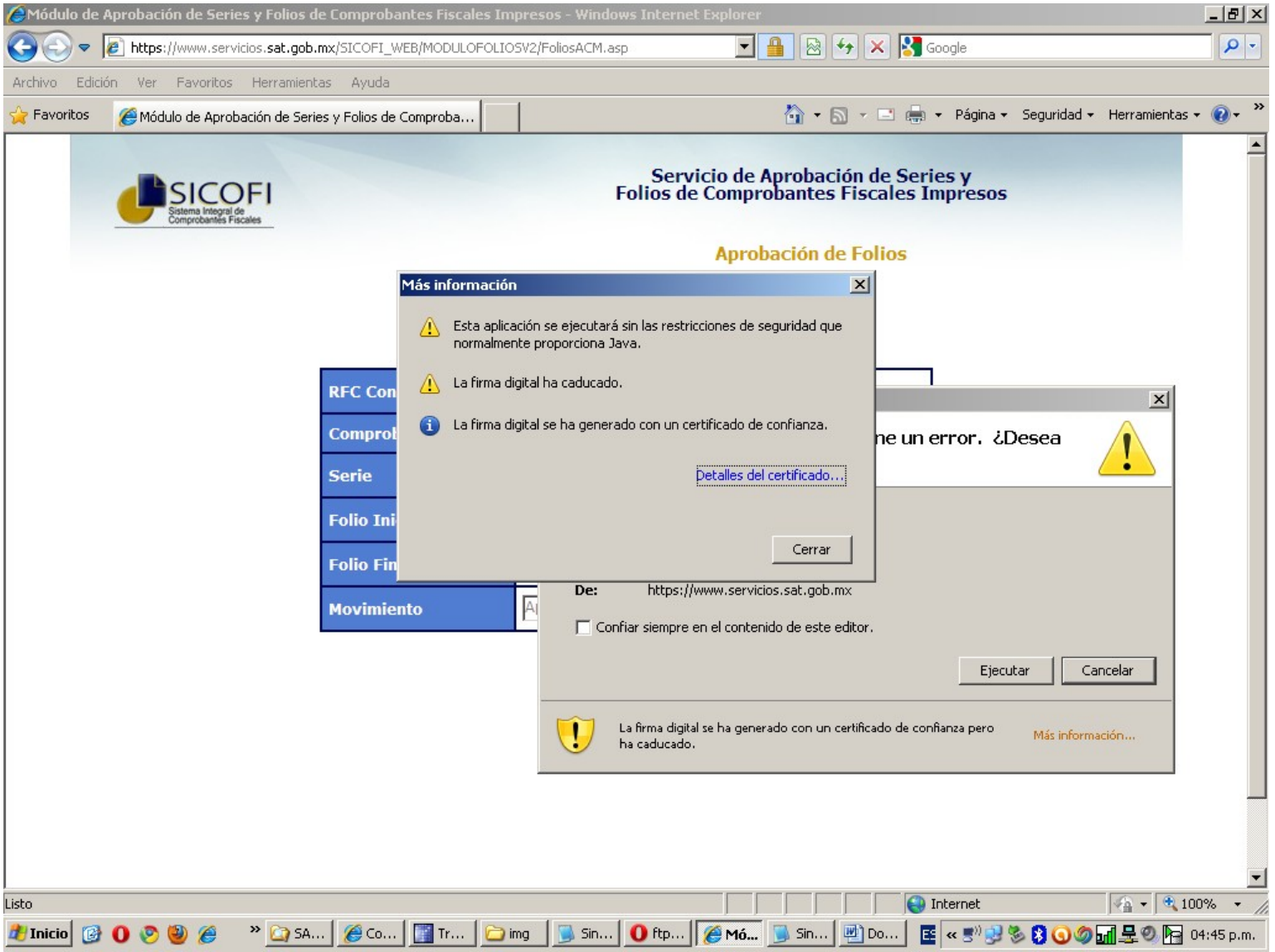
Problema operativo importante:

- Evasión de responsabilidades entre áreas de desarrollo y liberación a producción.

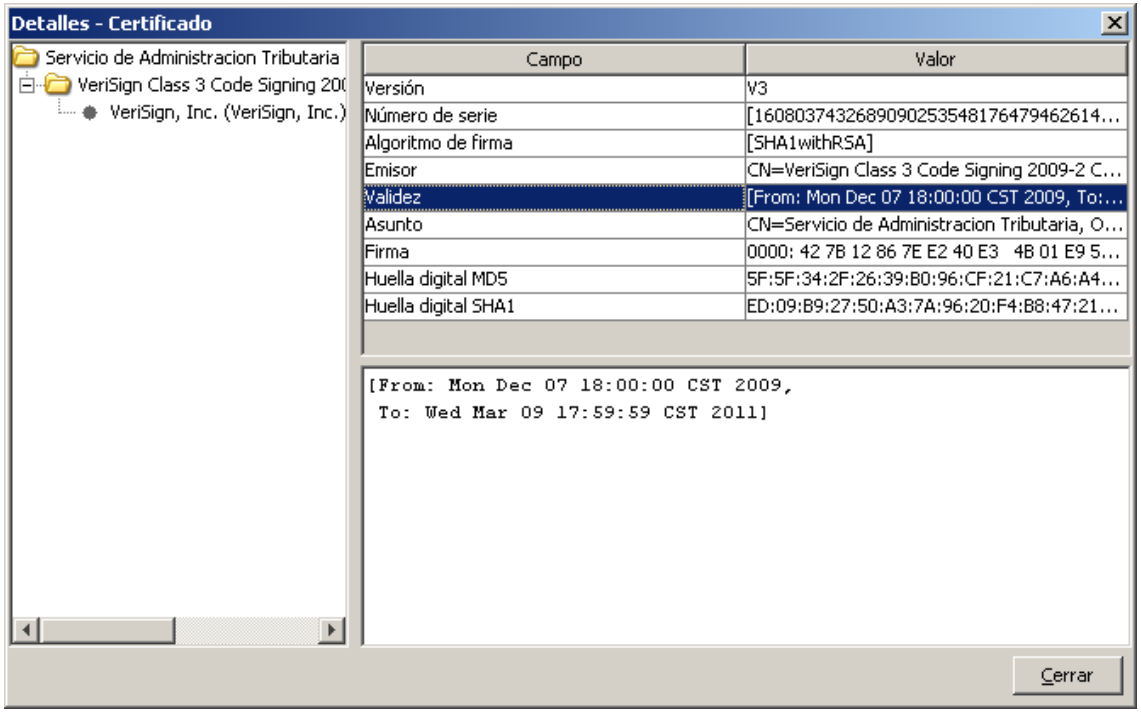
Ejemplo


Módulo de Aprobación de Series y Folios de Comprobantes Fiscales Impresos - Windows Internet Explorer
https://www.servicios.sat.gob.mx/SICOFI_WEB/MODULOFOLIOSV2/FoliosACM.asp
Archivo Edición Ver Favoritos Herramientas Ayuda
Favoritos
Módulo de Aprobación de Series y Folios de Comproba...
Página Seguridad Herramientas
SICOFI
Sistema Integral de Comprobantes Fiscales
Servicio de Aprobación de Series y Folios de Comprobantes Fiscales Impresos
Aprobación de Folios
DATOS DEL CONTRIBUYENTE
RFC Contribuyente
Comprobante
Serie
Folio Inicial
Folio Final
Movimiento
Advertencia - Seguridad
La firma digital de la aplicación tiene un error. ¿Desea ejecutar la aplicación?
Nombre: Applet1
Editor: Servicio de Administracion Tributaria
De: https://www.servicios.sat.gob.mx
Confiar siempre en el contenido de este editor.
Ejecutar
Cancelar
La firma digital se ha generado con un certificado de confianza pero ha caducado.
Más información...
Listo
Internet
100%
Inicio
04:44 p.m.

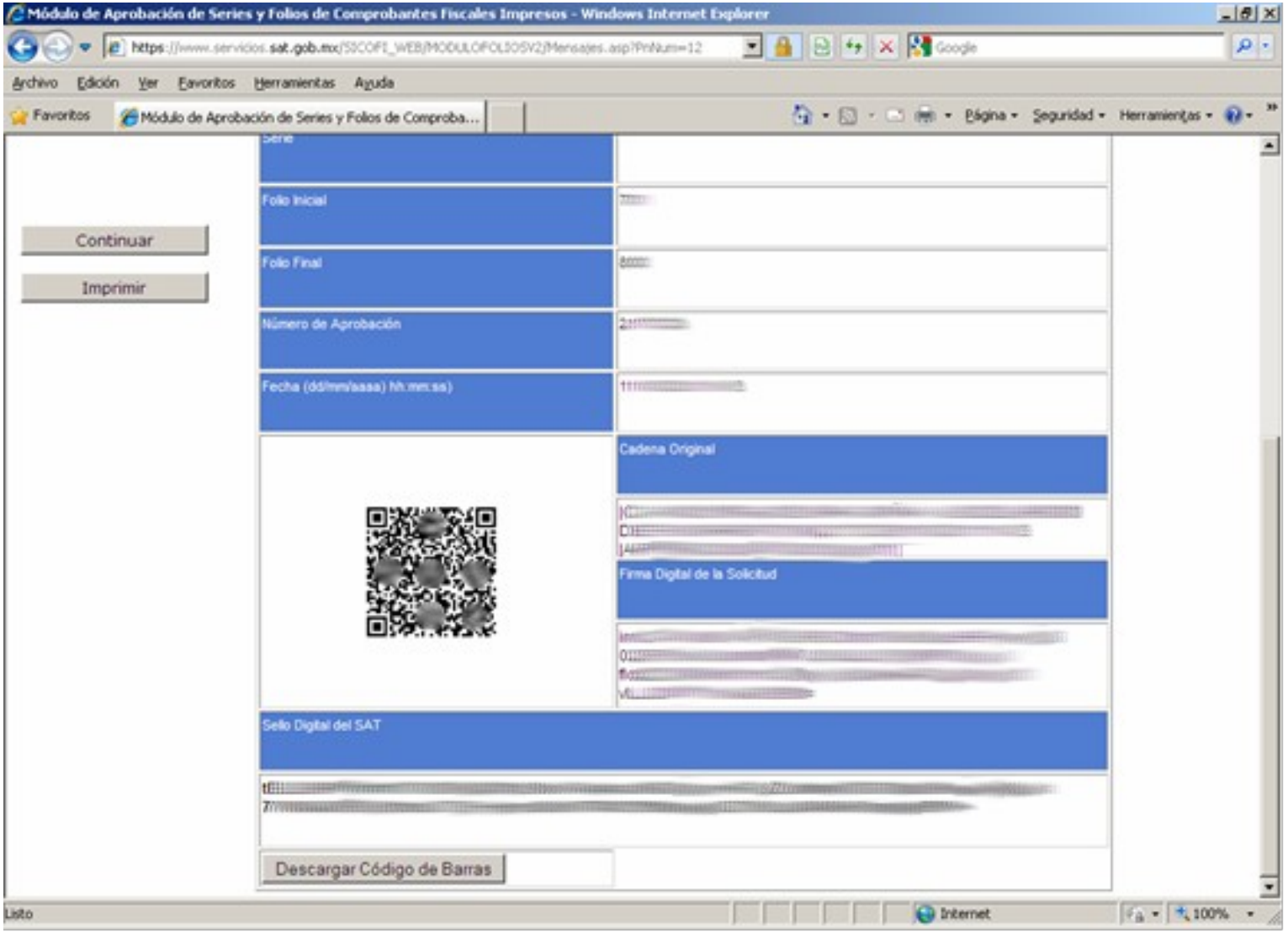
Ejemplo


Módulo de Aprobación de Series y Folios de Comprobantes Fiscales Impresos - Windows Internet Explorer
https://www.servicios.sat.gob.mx/SICOFI_WEB/MODULOFOLIOSV2/FoliosACM.asp
Archivo Edición Ver Favoritos Herramientas Ayuda
Favoritos
Módulo de Aprobación de Series y Folios de Comproba...
Página Seguridad Herramientas
SICOFI
Sistema Integral de Comprobantes Fiscales
Servicio de Aprobación de Series y Folios de Comprobantes Fiscales Impresos
Aprobación de Folios
Más información
Esta aplicación se ejecutará sin las restricciones de seguridad que normalmente proporciona Java.
La firma digital ha caducado.
La firma digital se ha generado con un certificado de confianza.
Detalles del certificado...
Cerrar
RFC Con
Comprol
Serie
Folio Ini
Folio Fin
Movimiento
ne un error. ¿Desea
De: https://www.servicios.sat.gob.mx
Confiar siempre en el contenido de este editor.
Ejecutar
Cancelar
La firma digital se ha generado con un certificado de confianza pero ha caducado.
Más información...
Listo
Internet
100%
Inicio
04:45 p.m.

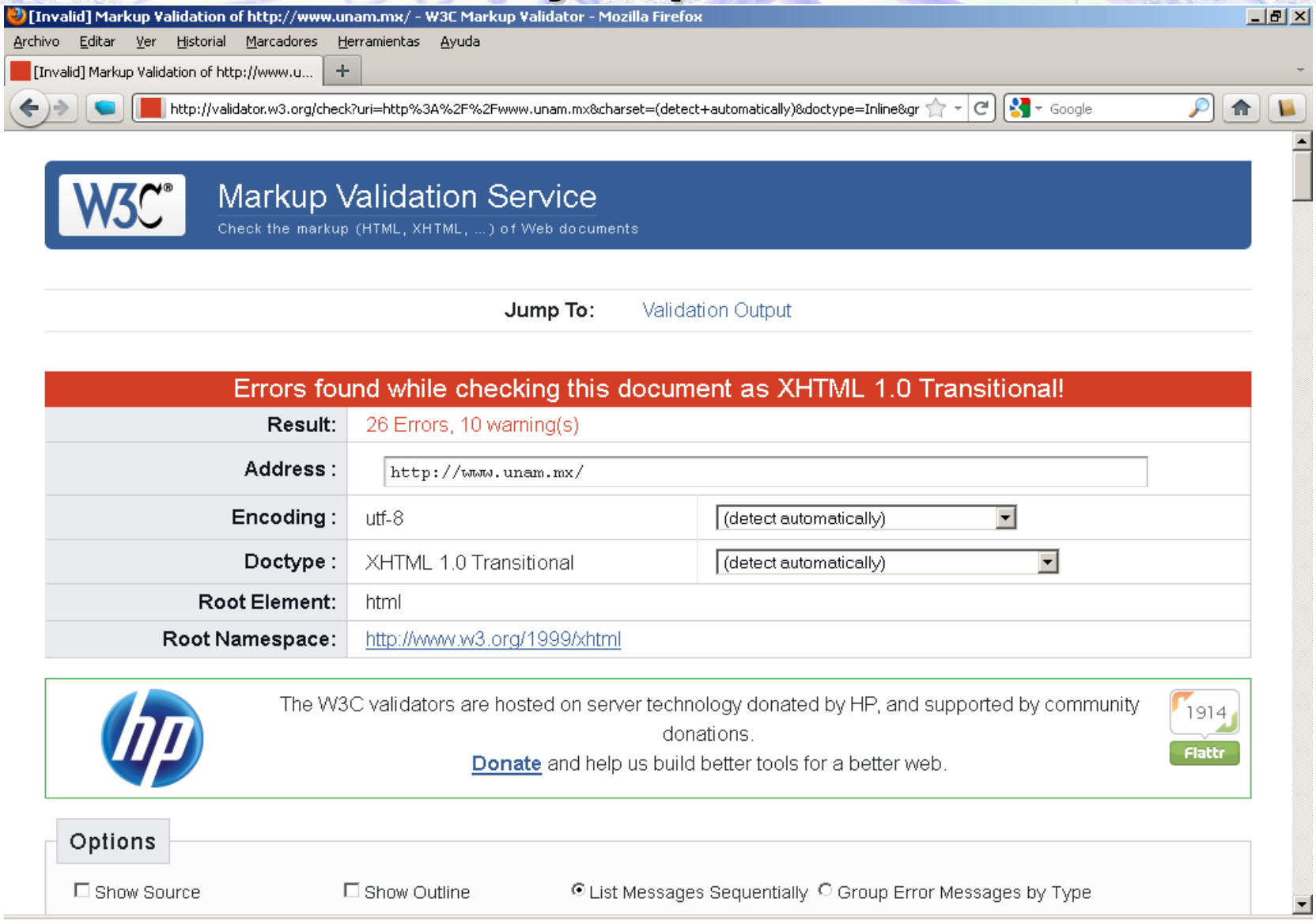
Ejemplo


Detalles - Certificado
Servicio de Administracion Tributaria
VeriSign Class 3 Code Signing 20(
VeriSign, Inc. (VeriSign, Inc.)
Campo
Valor
Versión
V3
Número de serie
[160803743268909025354817647946261 4...
Algoritmo de firma
[SHA1withRSA]
Emisor
CN=VeriSign Class 3 Code Signing 2009-2 C...
Validez
[From: Mon Dec 07 18:00:00 CST 2009, To:...
Asunto
CN=Servicio de Administracion Tributaria, O...
Firma
0000: 42 7B 12 86 7E E2 40 E3 4B 01 E9 5...
Huella digital MD5
5F:5F:34:2F:26:39:B0:96:CF:21:C7:A6:A4...
Huella digital SHA1
ED:09:B9:27:50:A3:7A:96:20:F4:B8:47:21...
[From: Mon Dec 07 18:00:00 CST 2009,
To: Wed Mar 09 17:59:59 CST 2011]
Cerrar

Ejemplo

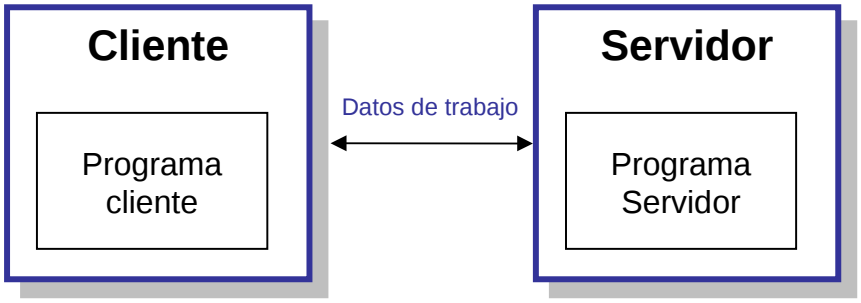

Módulo de Aprobación de Series y Folios de Comprobantes Fiscales Impresos - Windows Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Favoritos
Página Seguridad Herramientas
Folio Inicial
Folio Final
Número de Aprobación
Fecha (dd/mm/aaaa) hh:mm:ss)
Cadena Original
Firma Digital de la Solicitud
Sello Digital del SAT
Continuar
Imprimir
Descargar Código de Barras
Listo
Internet
100%

Ejemplo

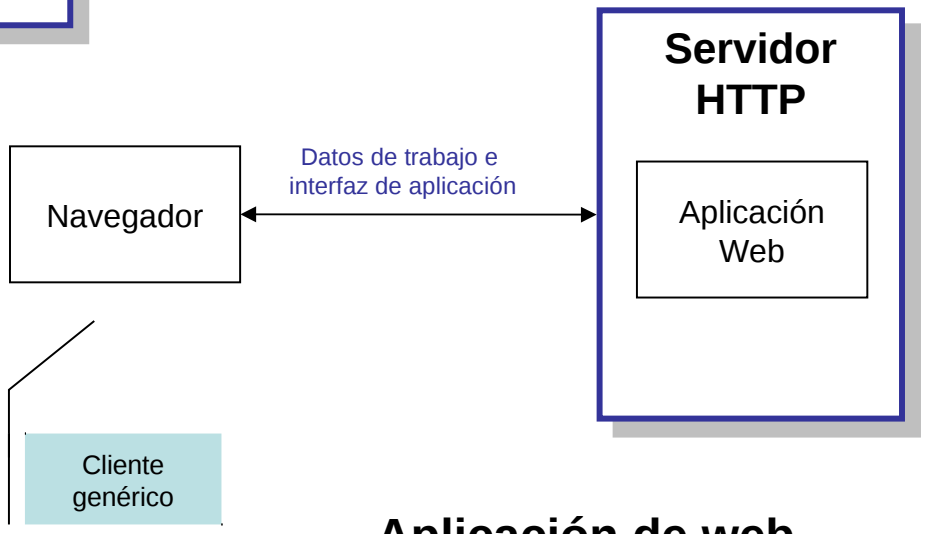

[Invalid] Markup Validation of http://www.unam.mx/ - W3C Markup Validator - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
[Invalid] Markup Validation of http://www.u...
http://validator.w3.org/check?uri=http%3A%2F%2Fwww.unam.mx&charset=(detect+automatically)&doctype=Inline&gr
Google
Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents
Jump To: Validation Output
Errors found while checking this document as XHTML 1.0 Transitional!
Result: 26 Errors, 10 warning(s)
Address : http://www.unam.mx/
Encoding : utf-8 (detect automatically)
Doctype : XHTML 1.0 Transitional (detect automatically)
Root Element: html
Root Namespace: http://www.w3.org/1999/xhtml
The W3C validators are hosted on server technology donated by HP, and supported by community donations.
Donate and help us build better tools for a better web.
1914
Flattr
Options
Show Source
Show Outline
List Messages Sequentially
Group Error Messages by Type

Ejemplo:
Aplicaciones de web

Aplicaciones de web



Aplicación Cliente-Servidor tradicional



Aplicación de web

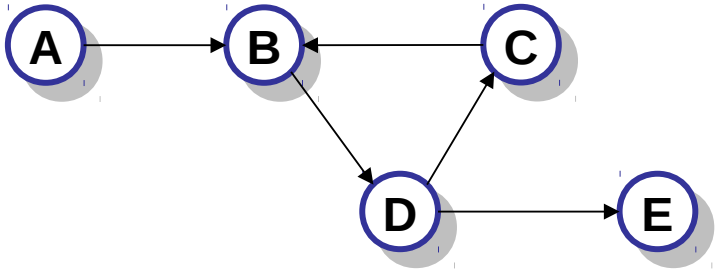
Aplicaciones de web

Diferencias principales entre
aplicaciones tradicionales y aplicaciones de web:

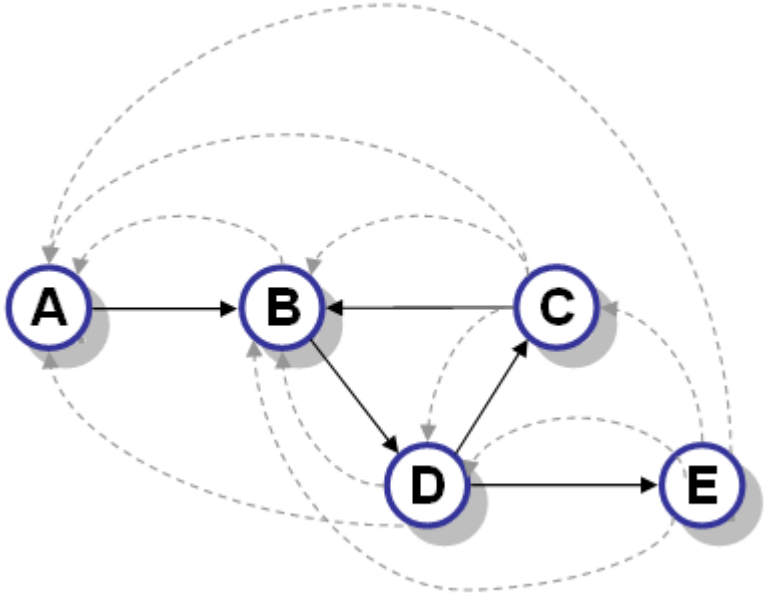
- Control de flujo
- Desacoplamiento del estado

Aplicaciones de web

Control de flujo



Aplicación tradicional



Aplicación de web

Aplicaciones de web

Desacoplamiento del estado

- Los servidores Web no conservan el estado.
- Los mensajes entre cliente y servidor son independientes.

Solución:

Mantener el estado

- en la lógica de la aplicación web.
- en el cliente (no recomendable)

Diversidad de productos de software

- **Servidores de web:**

- Apache HTTP Server
- Apache Tomcat
- Microsoft Internet Information Server
- Oracle WebLogic Server
- Sun Java System Web Server IBM HTTP Server
- Zeus Web Server
- …

- **Clientes de web:**

- Mozilla:
  - Mozilla
  - Firefox
  - Camino
  - SeaMonkey
  - Netscape
- Microsoft Internet Explorer
- Apple Safari
- Opera
- Google Chrome
- …

Tecnologías de servidor para la programación de aplicaciones

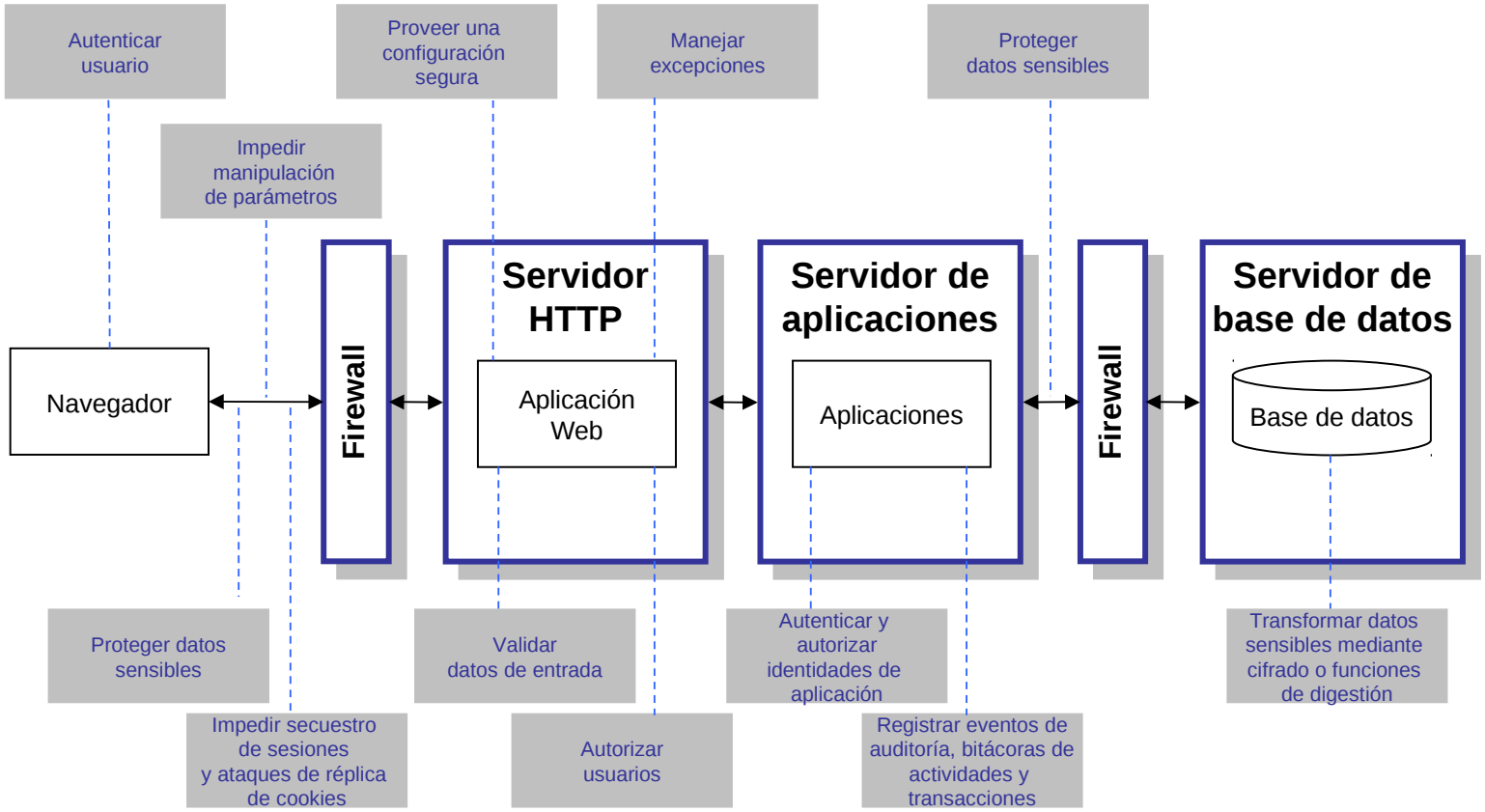
- CGI
- Servlets
- ASP, JSP, PHP
- Java Struts / JSF
- Web Services, SOAP
- J2EE, ASP.NET
- Web 2.0 / AJAX

Tecnologías de cliente

- Lenguajes de scripting:
  - JavaScript (Netscape/Mozilla)
  - JScript (Microsoft)
  - ECMAScript (European Computer Manufacturers Association)
  - Visual Basic Script (Microsoft)
- Java applets
- COM, ActiveX, .NET
- Flash, Shockwave
- Plugins (Acrobat, Quicktime, Real Player, Windows Media Player, etc.)
- XUL, XAML

Introducción a la seguridad de software

Ejemplo: Seguridad de aplicaciones de web



Introducción a la seguridad de software

Software seguro:

- Software que no puede ser subvertido o forzado a fallar intencionalmente. En resumen, software que continúa operando de manera correcta y predecible a pesar de esfuerzos deliberados para comprometer su fiabilidad.

Introducción a la seguridad de software

El software seguro ha sido diseñado, implementado, configurado y mantenido de manera que es capaz de:

- Continuar funcionando correctamente en presencia de la mayoría de los ataques ya sea por resistencia a la explotación de las fallas u otras debilidades o por tolerancia a errores y fallas que resultan de tales explotaciones.

- Aislar, contener y limitar los daños resultantes de fallas generadas por ataques que el software fue incapaz de resistir o tolerar, y recuperarse lo más rápidamente posible de dichas fallas.

Introducción a la seguridad de software

Software seguro:

El objetivo del aseguramiento de software es establecer una base para obtener confianza justificable de que tendrá las siguientes propiedades:

- Calidad
- Fiabilidad
- Correctez
- Usabilidad
- Interoperabilidad
- Consistencia
- Protección de recursos
- Tolerancia a fallas
- Trazabilidad

Modelos de referencia de seguridad en software

Principios de diseño y objetivos para el software seguro

1. Mínimo privilegio

   Cada entidad (usuario, proceso o dispositivo) recibe el conjunto más restrictivo de los privilegios necesarios para el desempeño de sus tareas autorizadas.

2. Mediación completa

   Para cada entidad que solicita el acceso a un recurso o la ejecución de un proceso, se debe ser verificar que tenga la debida autorización antes de conceder el acceso o permitir la ejecución. La solicitud debe ser rechazada si el solicitante no está debidamente autorizado.

3. Valores predeterminados a prueba de errores

   En toma de decisiones importantes de seguridad, por omisión, se debe denegar en lugar de permitir.

4. Separación de privilegios

   El acceso a recursos o procesos críticos no se debe permitir con base únicamente en una condición simple.

Modelos de referencia de seguridad en software

Principios de diseño y objetivos para el software seguro

5. Diseño abierto

   La seguridad de un mecanismo no debe depender de la secrecía de su diseño o implementación.

6. Trazabilidad

   Se deben generar registros de la actividad del software que puedan ayudar a determinar el comportamiento o el compromiso resultado de un ataque, o a entender el patrón de ataque, con el fin de resistir o tolerarlo de mejor manera. Después del compromiso, las trazas pueden ayudar a la recuperación, diagnóstico, reparación, análisis forense y rendición de cuentas.

7. Defensa en profundidad

   Cada una de las piezas o capas que constituyen el software tiene mecanismos de protección propios, independientes de aquellos del resto de los componentes

8. Proporcionalidad (Factor de trabajo).

   El costo de una contramedida o técnica de mitigación de una vulnerabilidad o debilidad debe ser proporcional al valor del activo relacionado.

Modelos de referencia de seguridad en software

Principios de diseño y objetivos para el software seguro

9. Economía de mecanismos

Los mecanismos de seguridad deben ser tan simples como sea posible.

10. Analizabilidad

El comportamiento del software debe ser analizable y predecible a partir de las descripciones de ingeniería, tales como las especificaciones de diseño y la codificación.

11. Aceptabilidad psicológica

Los mecanismos de seguridad no deben hacer que el acceso a los recursos sea más difícil que si estos mecanismos no estuviesen presentes.

Vulnerabilidades, amenazas y ataques al software

La seguridad en software:
¿un oxímoron?

Vulnerabilidades en software

Definiciones de vulnerabilidad:

- ISO 27005:

  Debilidad de un activo o grupo de activos que pueden ser explotados por una o más amenazas.

  > donde un activo es algo que puede tiene un valor para la organización, sus operaciones de negocio y la continuidad del mismo, incluidos los recursos de información que apoyan la misión de la organización.

- Committee on National Security Systems (CNSS):

  Debilidad en un sistema de información, en procedimientos de seguridad, controles internos o implementación del sistema que podrían ser explotados.
  (National Information Assurance Glossary - http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf)

Vulnerabilidades en software

Clasificaciones de vulnerabilidades: (1/2)

Fase del Ciclo de vida de Desarrollo de Software (SDLC - Software Development LifeCycle)

- *Diseño*
  - *1. Estudio de factibilidad*
  - *2. Definición de requerimientos*
  - *3. Diseño*
- *Implementación*
  - *4. Implementación*
  - *5. integración y pruebas*
- *Operaciones y mantenimiento*
  - *6. Operaciones y mantenimiento*

- *Génesis (origen)*
  - *Accidental*
  - *Intencional*
    - *Malicioso*
    - *No malicioso*

- *Ubicación según arquitectura*
  - *Capa de red*
  - *Sistema operativo*

Vulnerabilidades en software

Clasificaciones de vulnerabilidades: (2/2)

- *Tecnología afectada*
  - *Desbordamiento de buffer en C*
  - *XSS en aplicación web*
  - *SYN flood en TCP*
- *Errores o malas prácticas*
  - *Uso de contraseñas débiles*
  - *Falta de protección de datos sensibles*
- *Escenario de ataque habilitado*
  - *Comunicación de texto claro*
  - *Manejo inseguro de protocolos de estado*
  - *Incapacidad de manejar flujos anormales de paquetes*
- *Proceso de divulgación (disclosure)*
  - *Día-cero*
- *Explotabilidad*

Vulnerabilidades en software

Clasificación CLASP
(Comprehensive, Lightweight Application Security Process)

- Utiliza una clasificación enfocada en la enumeración de errores, también incluye las condiciones resultantes de los errores, así como eventos
- Tiene las siguientes categorías en el nivel superior:
  - Errores de rango y tipo.
    - Condición “escribir-que-donde” (“write-what-where”)“
    - desbordamiento de búfer
    - Problemas de cadena de formato.
  - Problemas Ambientales.
    - falta de un generador de números aleatorios.
  - Errores de sincronización y de manejo de tiempo.
    - vulnerabilidad “captura-repetición“
  - Errores de protocolo.
    - Uso de un algoritmo criptográfico débil o riesgoso.
  - Errores lógicos generales.
    - Incluye vulnerabilidades diversas como uso de un generador "no criptográfico" de números aleatorios, o una función que recibe muy pocos parámetros (por ej., el problema de cadena de formato en "C").

Vulnerabilidades en software

Clasificación de Siete reinos (Seven Kingdoms)

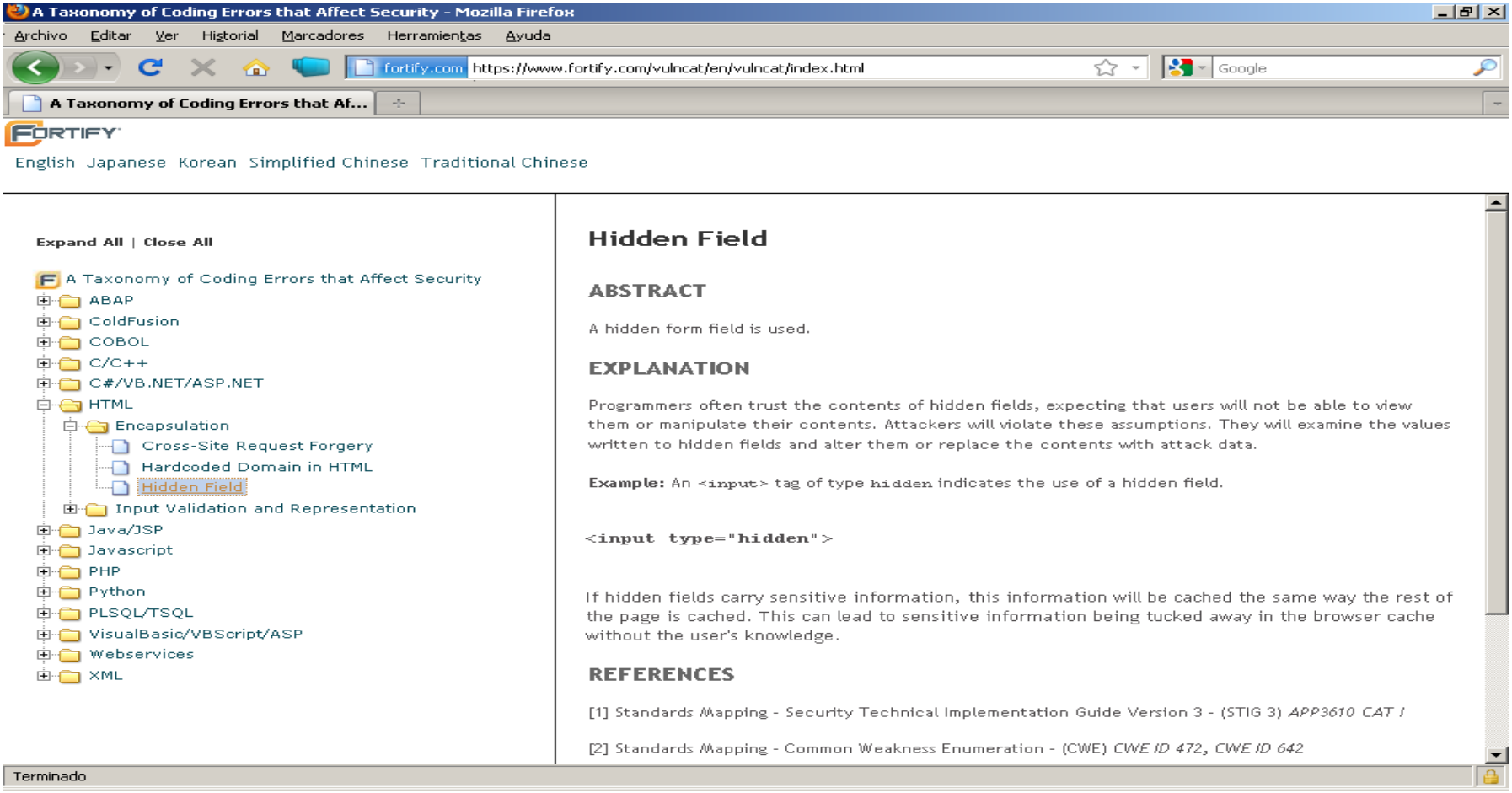
Esta clasificación de errores de seguridad de software tiene los siguientes ocho niveles superiores:

1. Validación y representación de datos de entrada
2. Abuso de API
3. Características de seguridad
4. Tiempo y estado
5. Manejo de errores
6. Calidad de código
7. Encapsulación
8. Medio ambiente (*)

* Esta categoría se compone mayormente de problemas de configuración (vulnerabilidades que no tienen cabida en los primeros 7 niveles)

Vulnerabilidades en software

Registro de vulnerabilidades en lenguajes de programación (según clasificación de Siete Reinos)
https://www.fortify.com/vulncat



Vulnerabilidades en software

Registro de vulnerabilidades:

- National Institute of Standards and Technology (NIST)

  National Vulnerability Database (NVD)
  http://nvd.nist.gov

  NVD Common Vulnerability Scoring System
  http://nvd.nist.gov/cvss.cfm

- US Computer Emergency Response Team (US-CERT) Vulnerability Notes Database
  http://www.kb.cert.org/vuls

- MITRE Common Vulnerabilities and Exposures (CVE)
  http://cve.mitre.org

- Open Source Vulnerability Database
  http://osvdb.org

- eEye Research Zero-Day Tracker
  http://research.eeye.com/html/alerts/zeroday

Vulnerabilidades en software

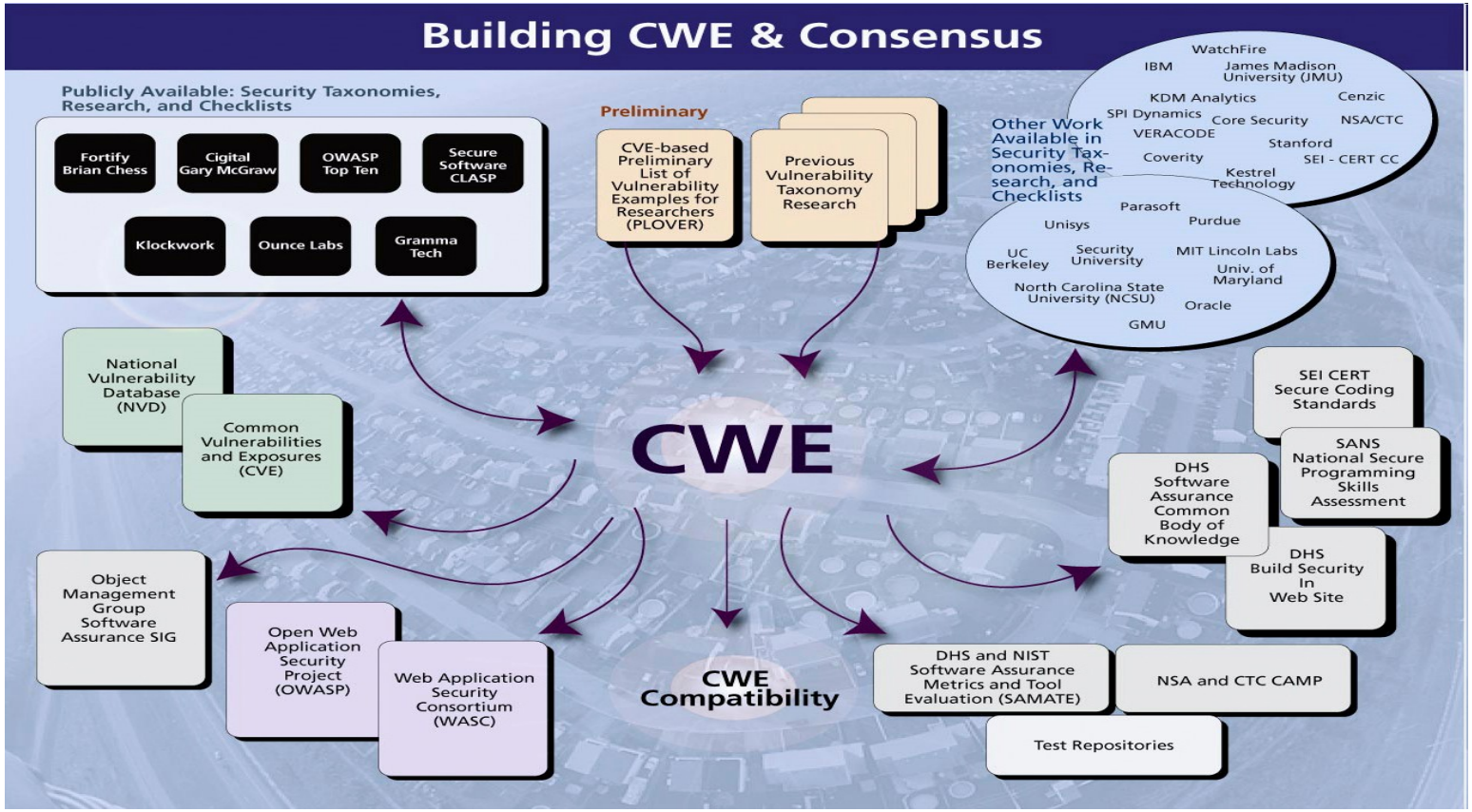
Análisis de vulnerabilidades
(Clasificación por importancia)

- SANS (SysAdmin, Audit, Network, Security) Institute
  http://www.sans.org
- The MITRE Corporation
  http://www.mitre.org
- The Open Web Application Security Project
  http://www.owasp.org
- Web Application Security Consortium
  http://www.webappsec.org

Vulnerabilidades en software

Common Weakness Enumeration (CWE)

http://cwe.mitre.org

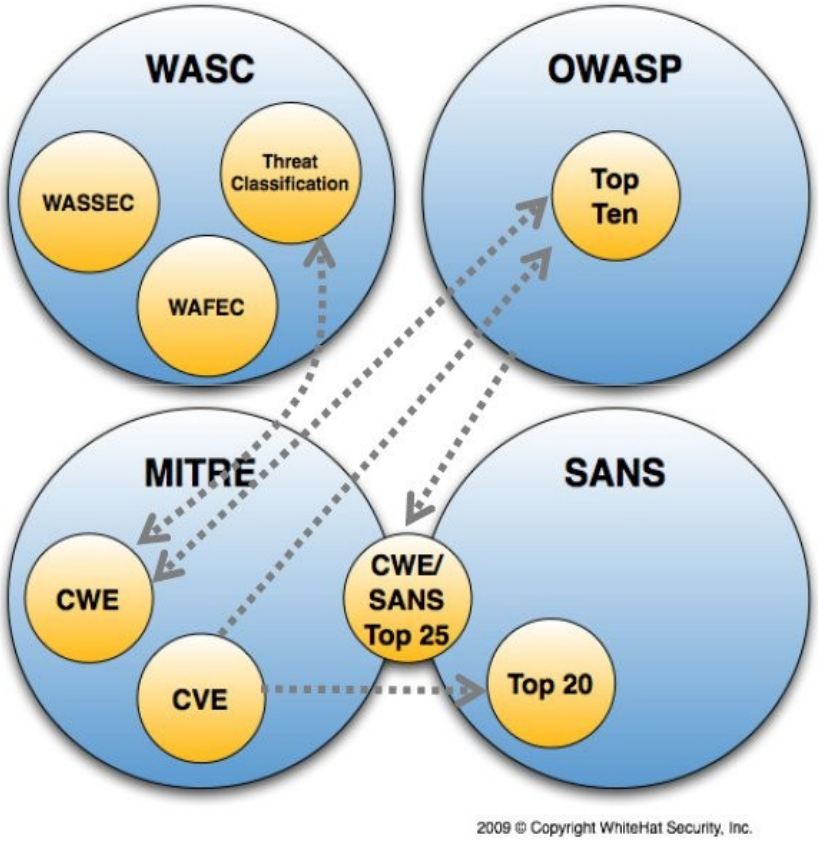


Vulnerabilidades en software

Principales vulnerabilidades:

- CWE/SANS Top 25 Most Dangerous Software Errors - 2010
  http://cwe.mitre.org/top25

- OWASP Top 10 Most Critical Web Application Security Risks - 2010
  http://www.owasp.org/index.php/Top_10_2010



Vulnerabilidades en software

CWE/SANS Top 25 Most Dangerous Software Errors – 2010 (1/2)

Rango	Calificación	Id	Nombre
1	346	CWE-79	Neutralización inadecuada de datos de entrada durante la generación de página web ("Cross-site Scripting")
2	330	CWE-89	Neutralización inadecuada de elementos especiales utilizados en comando SQL ('SQL Injection')
3	273	CWE-120	Copia de búfer sin comprobación del tamaño de entrada ("desbordamiento de búfer clásico")
4	261	CWE-352	Falsificación de peticiones hacia sitios web (CSRF)
5	219	CWE-285	Control de acceso inadecuado (autorización)
6	202	CWE-807	Dependencia de entradas no confiables en una decisión de seguridad
7	197	CWE-22	Limitación inadecuada de nombre de ruta a un directorio restringido ('Recorrido de directorios")
8	194	CWE-434	Carga sin restricciones de archivo con extensión (tipo) peligroso
9	188	CWE-78	Neutralización inadecuada de elementos especiales utilizados en comando de sistema operativo ("Inyección de comando de SO")
10	188	CWE-311	Falta de cifrado de datos sensibles
11	176	CWE-798	Uso de credenciales en código duro
12	158	CWE-805	Acceso a búfer con valor de longitud incorrecto

Vulnerabilidades en software

CWE/SANS Top 25 Most Dangerous Software Errors – 2010 (2/2)

Rango	Calificación	Id	Nombre
13	157	CWE-98	Control inadecuado del nombre de archivo que incluye/requiere instrucciones en programa PHP ("Inclusión de archivos PHP)
14	156	CWE-129	Validación incorrecta de índice de arreglo
15	155	CWE-754	Comprobación inadecuada de condiciones inusuales o excepcionales
16	154	CWE-209	Exposición de información a través de un mensaje de error
17	154	CWE-190	Desbordamiento o mapeo circular de enteros
18	153	CWE-131	Cálculo incorrecto de tamaño de búfer
19	147	CWE-306	Falta de autenticación para función crítica
20	146	CWE-494	Descarga de código sin control de integridad
21	145	CWE-732	Asignación incorrecta de permisos para recursos críticos
22	145	CWE-770	Asignación ilimitada de recursos
23	142	CWE-601	Redireccionamiento de URL hacia sitio que no es de confianza ("Redireccionamiento abierto")
24	141	CWE-327	Uso de algoritmo de cifrado débil o riesgoso
25	138	CWE-362	Condición de carrera

Vulnerabilidades en software

Riesgos críticos de seguridad de aplicaciones Web (10 más importantes)

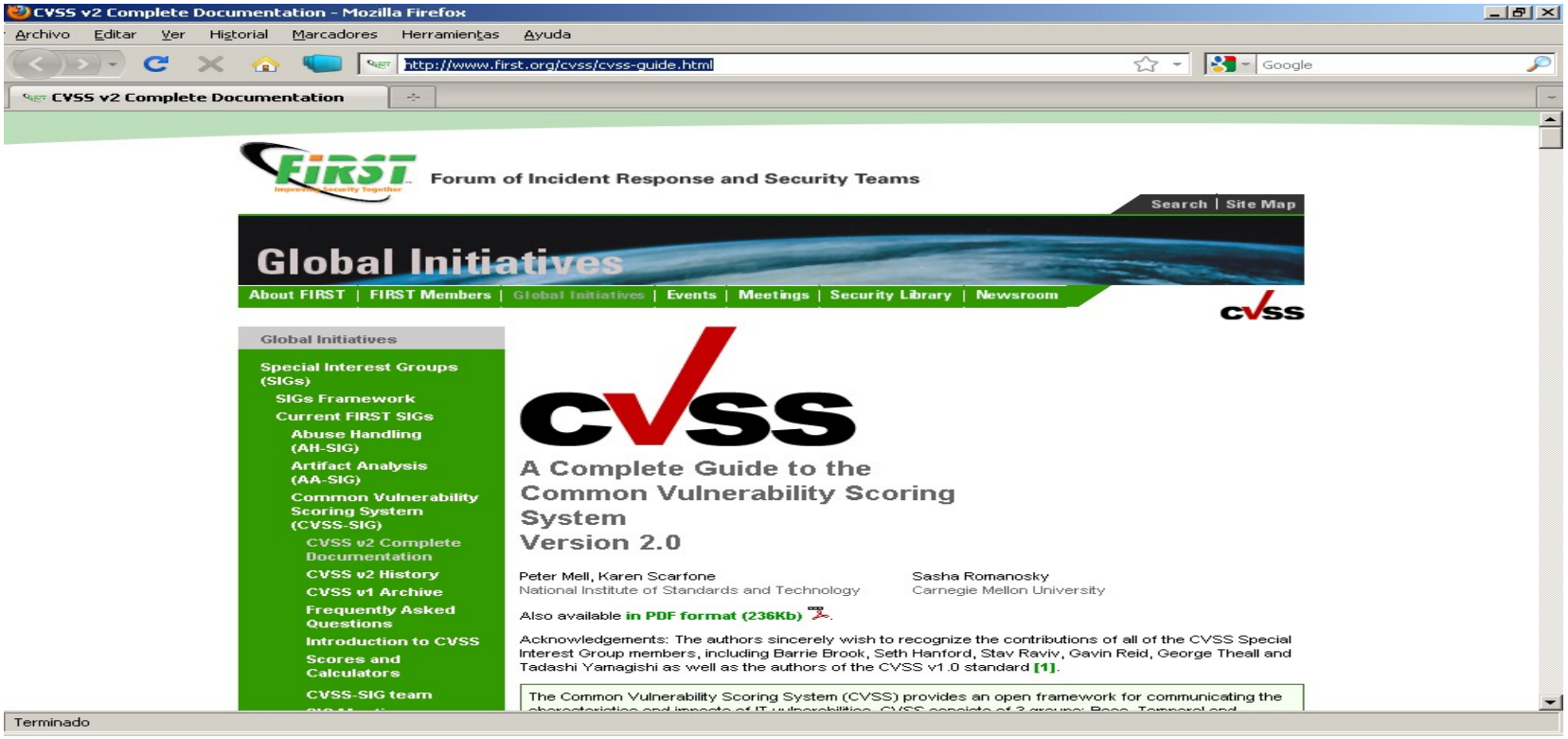
OWASP Top 10 – 2010		OWASP Top 10 – 2007	
A1	Inyección	A2	Fallas de inyección
A2	Cross-Site Scripting (XSS)	A1	Cross-Site Scripting (XSS)
A3	Autenticación y administración incorrectas de sesiones	A7	Autenticación y administración incorrectas de sesiones
A4	Referencias directas inseguras hacia objetos	A4	Referencias directas inseguras hacia objetos
A5	Falsificación de peticiones al servidor (CSRF)	A5	Falsificación de peticiones al servidor (CSRF)
A6	Configuración incorrecta de seguridad (nuevo)		(fue A10 en Top 10 2004) Administración insegura de configuración
A7	Almacenamiento criptográfico inseguro	A8	Almacenamiento criptográfico inseguro
A8	Falla en restricción de acceso a URL	A10	Falla en restricción de acceso a URL
A9	Protección insuficiente de la capa de transporte	A9	Comunicaciones inseguras
A10	Transiciones internas y externas no validadas (nuevo)		(No presente en Top 10 2007)
	eliminado en 2010	A3	Ejecución maliciosa de archivo
	eliminado en 2010	A6	Fuga de información y administración inadecuada de errores

Vulnerabilidades en software

Medición y calificación de vulnerabilidades

Forum of Incident Response and Security Teams (FIRST)
Common Vulnerability Scoring System

http://www.first.org/cvss/cvss-guide.html



Amenazas y ataques al software

- Amenaza

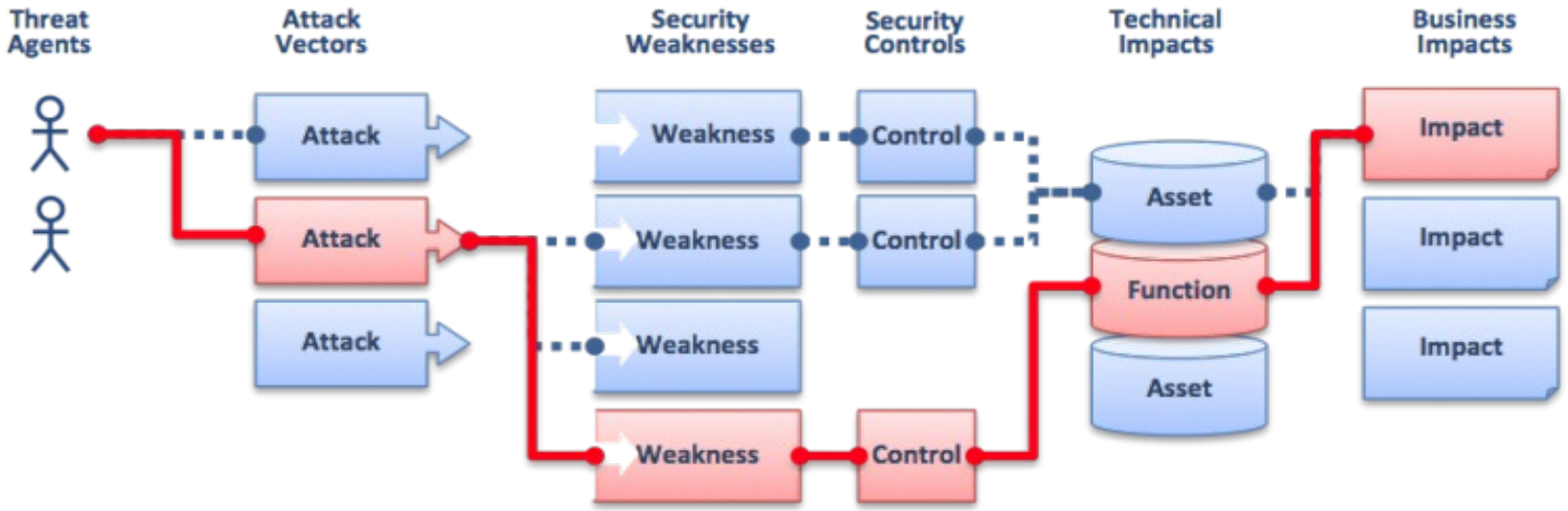
  Cualquier circunstancia o evento con potencial de impactar negativamente en las operaciones de la organización (incluyendo misión, funciones, imagen o reputación), los activos e individuos de la organización, u otras organizaciones.

Categoría de amenaza	Descripción	Propiedad comprometida	Objetivos
Sabotaje	La ejecución del software o se suspende, se termina o se degrada su rendimiento o bien el ejecutable se borra o destruye.	Disponibilidad	* DoS
Subversión	El software ejecutable se modifica intencionalmente (alterado maliciosamente o corrompido) o es substituido por un usuario no autorizado o una lógica no autorizada insertada en el ejecutable (la mayoría de las veces código malicioso).	Integridad	* Transformación del software en un puente para ser utilizado como elemento de ataque * Impedir que el software realice sus funciones previstas correctamente o de manera predecible (una forma de sabotaje así como de subversión)
Intercepción	Una entidad no autorizada tiene acceso al software (o a una función restringida dentro de él).	Control de acceso	* Ejecución no autorizada * Copia ilegal o robo del ejecutable
Liberación de información	Los detalles tecnológicos y de implementación del software quedan expuestos mediante ingeniería inversa (por ej., decompilación o desensamble)	Confidencialidad	* Reconocimiento preataque * Obtención de propiedad intelectual

Amenazas y ataques al software

- Ataque

  Cualquier tipo de actividad maliciosa que intenta recolectar, interrumpir, denegar, degradar o destruir recursos del sistema de información o la información misma.



Amenazas y ataques al software

Patrón de ataque

- Modelo para un explotar una vulnerabilidad. Es una descripción de un enfoque común empleado por los atacantes tomar en sus ataques al software.
- Cada patrón se desarrolla mediante un análisis extenso de las características comunes de grandes conjuntos de ataques tanto exitosos como fallidos.
- Ayuda a identificar y calificar el riesgo que una determinada explotación producirá en un sistema de software

Amenazas y ataques al software

Patrones de ataque

Make the Client Invisible

Command Delimiters

Use a User-Supplied Configuration File to Run Commands That Elevate Privilege

Passing Local Filenames to Functions That Expect a URL

User-Supplied Variable Passed to File System Calls

Buffer Overflow in Local Command-Line Utilities

Using Escaped Slashes in Alternate Encoding

Client-Controlled Environment Variables

Direct Access to Executable Files

Embedding Scripts within Scripts

Argument Injection

Multiple Parsers and Double Escapes

Postfix NULL Terminator

Relative Path Traversal

Simple Script Injection

XSS in HTTP Headers

HTTP Query Strings

User-Controlled Filename

Using Slashes in Alternate Encoding

User-Supplied Global Variables (DEBUG=1, PHP Globals, and So Forth)

Target Programs That Write to Privileged OS Resources

Leverage Executable Code in Nonexecutable Files

File System Function Injection, Content Based

Buffer Overflow with Environment Variables

Postfix, Null Terminate, and Backslash

Meta-characters in E-mail Header

Client-side Injection, Buffer Overflow

Cause Web Server Misclassification

Unicode Encoding

UTF-8 Encoding

URL Encoding

Alternative IP Addresses

Slashes and URL Encoding Combined

.

Web Logs

MIME Conversion

Attack Pattern Fragment: Manipulating Terminal Devices

Analog In-Band Switching Signals (aka “Blue Boxing”)

Alternate Encoding the Leading Ghost Characters

Make Use of Configuration File Search Paths

Session ID, Resource ID, and Blind Trust

Embedding Script in Nonscript Elements

Overflow Binary Resource File

Overflow Variables and Tags

Overflow Symbolic Links

HTTP Cookies

Filter Failure through Buffer Overflow

Buffer Overflow in an API Call

Parameter Expansion

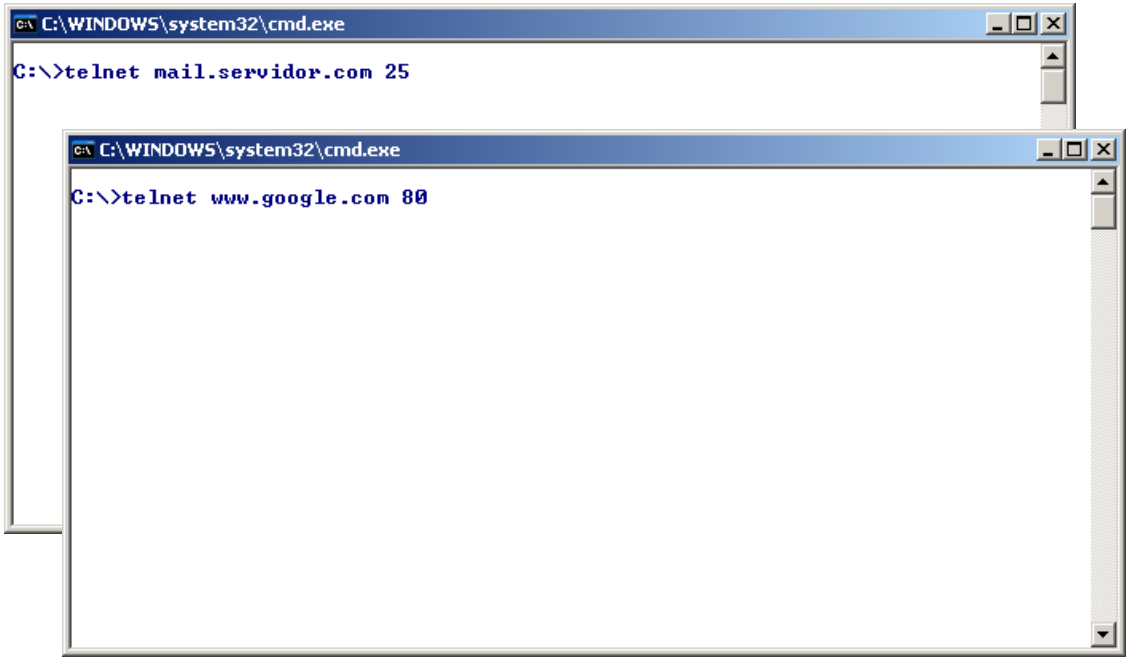
String Format Overflow in syslog()

.

Amenazas y ataques al software

- Ejemplo:
  Patrón de ataque 1 - Volver invisible al cliente:

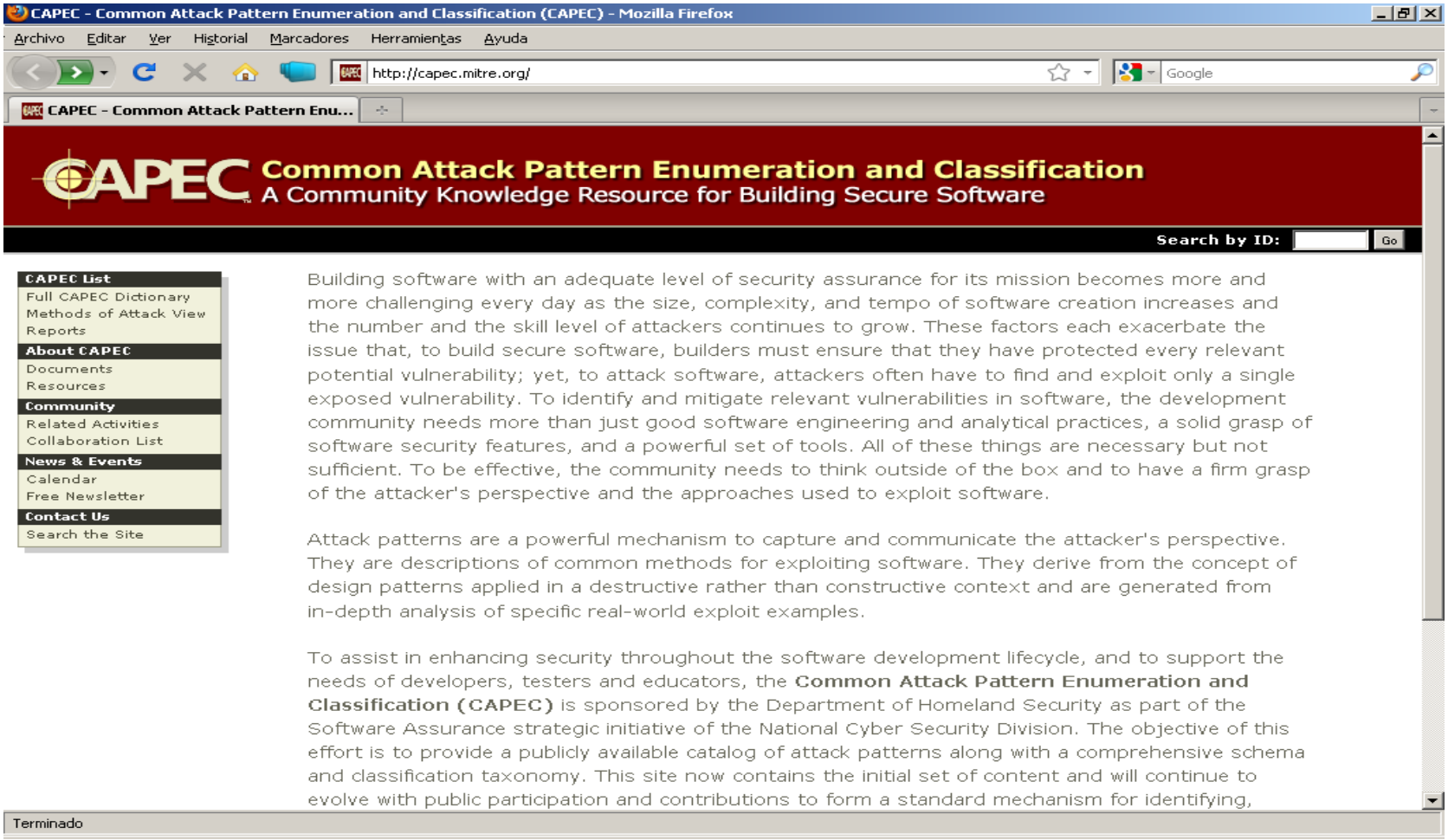
  - Eliminar al cliente de la sesión de comunicaciones y comunicarse directamente con el servidor
  - Tomar ventaja del modelo incorrecto de confianza



Amenazas y ataques al software

Common Attack Pattern Enumeration and Classification (CAPEC)

http://capec.mitre.org



Modelos de referencia de seguridad en software

La seguridad en software:
¿un oxímoron?

Modelos de referencia de seguridad en software

Principios de diseño y objetivos para el software seguro

1. Mínimo privilegio
2. Mediación completa
3. Valores predeterminados a prueba de errores
4. Separación de privilegios
5. Diseño abierto
6. Trazabilidad (registro de compromisos)
7. Defensa en profundidad
8. Proporcionalidad (Factor de trabajo).
9. Economía de mecanismos
10. Analizabilidad
11. Aceptabilidad psicológica

Modelos de referencia de seguridad en software

Systems Security Engineering - Capability Maturity Model (SSE-CMM)

http://www.sse-cmm.org

- Modelo de referencia de procesos que define los requerimientos para la implementación de seguridad dentro de uno o más sistemas relacionados.
  Es una compilación de las mejores prácticas conocidas para la ingeniería de seguridad.

- En un principio definido por la Agencia de Seguridad Nacional (NSA),
  ahora es un estándar internacional (ISO / IEC 21827).

- Permite a una organización de desarrollo de sistema añadir prácticas de seguridad en sus procesos CMM de ingeniería de sistemas (SE).
  SSE-CMM aumenta las áreas de proceso en el CMM tradicional mediante la incorporación de áreas relacionadas con ingeniería de seguridad.

- Contiene 129 prácticas base, organizadas en 22 áreas de proceso.
  Entre éstas existen 61 prácticas base, divididas en 11 áreas de proceso, encargadas del área de ingeniería de seguridad.

Modelos de referencia de seguridad en software

SSE-CMM – Niveles de madurez

Nivel 1 - Inicial: Este nivel se enfoca en la realización de las prácticas base de manera informal.

- 1.1 Se cumple con la realización de las prácticas base

Nivel 2 – Administrado: Se enfoca en los asuntos de definición, planeación y ejecución de un proyecto.

- 2.1 Planeación de la ejecución.
- 2.2 Ejecución disciplinada.
- 2.3 Verificación de la ejecución.
- 2.4 Monitoreo de la ejecución.

Nivel 3 – Definido: Se enfoca en una adaptación disciplinada de los procesos definidos a nivel de organización.

- 3.1 Definir un proceso estándar.
- 3.2 Realizar un proceso definido.
- 3.3 Coordinación del proceso

Nivel 4 – Cuantitativamente administrado: Su propósito es alinear las medidas del proceso con los objetivos propios del negocio.

- 4.1 Establecer objetivos de calidad medibles.
- 4.2 Gestión de la ejecución objetiva

Nivel 5 – En optimización: Mejora constante de las prácticas realizadas.

- 5.1 Mejoramiento de la capacidad organizacional.
- 5.2 Mejoramiento de la capacidad de los procesos

Modelos de referencia de seguridad en software

SSE-CMM

Áreas de proceso orientadas a la ingeniería de seguridad

- PA01 Administración de los controles de seguridad.
- PA02 Evaluación del impacto.
- PA03 Evaluación del riesgo de la seguridad.
- PA04 Evaluación de las amenazas.
- PA05 Evaluación de las vulnerabilidades.
- PA06 Construcción de un argumento de garantía.
- PA07 Coordinación de la seguridad.
- PA08 Monitoreo de la postura de seguridad.
- PA09 Proporcionar información de seguridad.
- PA10 Especificar necesidades de seguridad.
- PA11 Verificar y validar la seguridad

Áreas de proceso relacionadas con las prácticas organizacionales y de proyecto

- PA12 Garantizar la calidad.
- PA13 Administración de la configuración.
- PA14 Administración del riesgo del proyecto.
- PA15 Monitoreo y control del esfuerzo técnico.
- PA16 Planeación del esfuerzo técnico.
- PA17 Definir el proceso de ingeniería de sistemas de la organización.
- PA18 Mejora del proceso de ingeniería de sistemas de la organización.
- PA19 Administración de la evolución de la línea del producto.
- PA20 Administración del ambiente de soporte de la ingeniería de sistemas.
- PA21 Proveer conocimiento y habilidades continuas.
- PA22 Coordinación con proveedores.

Modelos de referencia de seguridad en software

Common Criteria

http://www.commoncriteriaportal.org

- Norma internacional (ISO / IEC 15408) para la certificación de seguridad informática (Common Criteria for Information Technology Security Evaluation)

- Marco de trabajo en el que los usuarios de computadoras pueden especificar sus requerimientos funcionales de seguridad, los fabricantes pueden implementar y/o aseverar acerca del cumplimiento sobre los atributos de seguridad de sus productos y los laboratorios pueden evaluar los productos para determinar si, efectivamente, cumplen dichos requerimientos.

- Ofrece certeza de que el proceso de especificación, implementación y evaluación de un producto de seguridad informática ha llevado a cabo de manera rigurosa y estándar.

- Define siete conjuntos predefinidos de requerimientos de seguridad conocidos como Evaluation Assurance Levels (EAL).

Modelos de referencia de seguridad en software

Common Criteria

Niveles de evaluación

- EAL1 Probado funcionalmente.
- EAL2 Probado estructuralmente.
- EAL3 Probado y verificado metodológicamente.
- EAL4 Diseñado, probado y revisado metodológicamente.
- EAL5 Diseñado y probado semi-formalmente.
- EAL6 Diseñado verificado semi-formalmente y probado.
- EAL7 Diseño verificado formalmente y probado.

Seguridad y SDLC (Software Development Life Cycle)

La seguridad en software:
¿un oxímoron?

Seguridad y SDLC

Actividades a incluir en SDLC

- **Requerimientos de seguridad:**
  - Se definen los requerimientos de seguridad imperativos al negocio, que pueden determinarse de acuerdo con regulaciones de la industria, políticas corporativas y necesidades específicas del negocio.

- **Arquitectura de seguridad:**
  - Se enfoca en la creación de una arquitectura de seguridad global. A partir de los requerimientos de seguridad determinados previamente se crea una propuesta de arquitectura de seguridad. Esta actividad clasifica las decisiones arquitectónicas del sistema a través de procesos de análisis de riesgos y de mitigación bien definidos.
  - Se identificarán los patrones de seguridad que permitirán cumplir con los requerimientos de seguridad y los puntualiza, indicando riesgos y vulnerabilidades conocidas.
  - Se crea una arquitectura prototipo y se perfecciona antes de que se comience con la siguiente actividad.
  - Es importante tomar en cuenta los requerimientos no-funcionales restantes con el objetivo de asegurarse que la arquitectura propuesta no los comprometerá.

Seguridad y SDLC

Actividades a incluir en SDLC

- **Diseño de seguridad:**
  - A partir de la arquitectura propuesta, se actualiza ésta utilizando métodos como análisis de factor, análisis de capas, políticas de seguridad, amenazas, clasificación de información y etiquetado. Se crea y documenta el diseño tomando en cuenta las mejores prácticas y los riesgos de cada uno de los patrones identificados.

- **Implementación de seguridad:**
  - Se lleva a la práctica el diseño de seguridad. La construcción del software debe seguir las mejores prácticas de código seguro.

- **Pruebas de caja negra:**
  - Se realizan pruebas de seguridad intentando romper el sistema. En este caso, las pruebas se realizan por personal que no tiene conocimiento alguno acerca de su implementación.

- **Pruebas de caja blanca:**
  - Se revisa el código y se buscan huecos de seguridad que pueden ser explotados. Se prueban diferentes ataques de seguridad orientados a comprometer el sistema, tratando de demostrar la manera en que se podría corromper los requerimientos de seguridad.

Seguridad y SDLC

Actividades a incluir en SDLC

- **Monitoreo:**
  - Es una actividad constante, se efectúa en todo momento mientras la aplicación está en producción y se debe realizarse no solo a nivel de aplicación; también deben incluirse todos los elementos de infraestructura relacionados. Deben quedar claramente identificados los eventos de seguridad respecto a los operativos o de negocio.

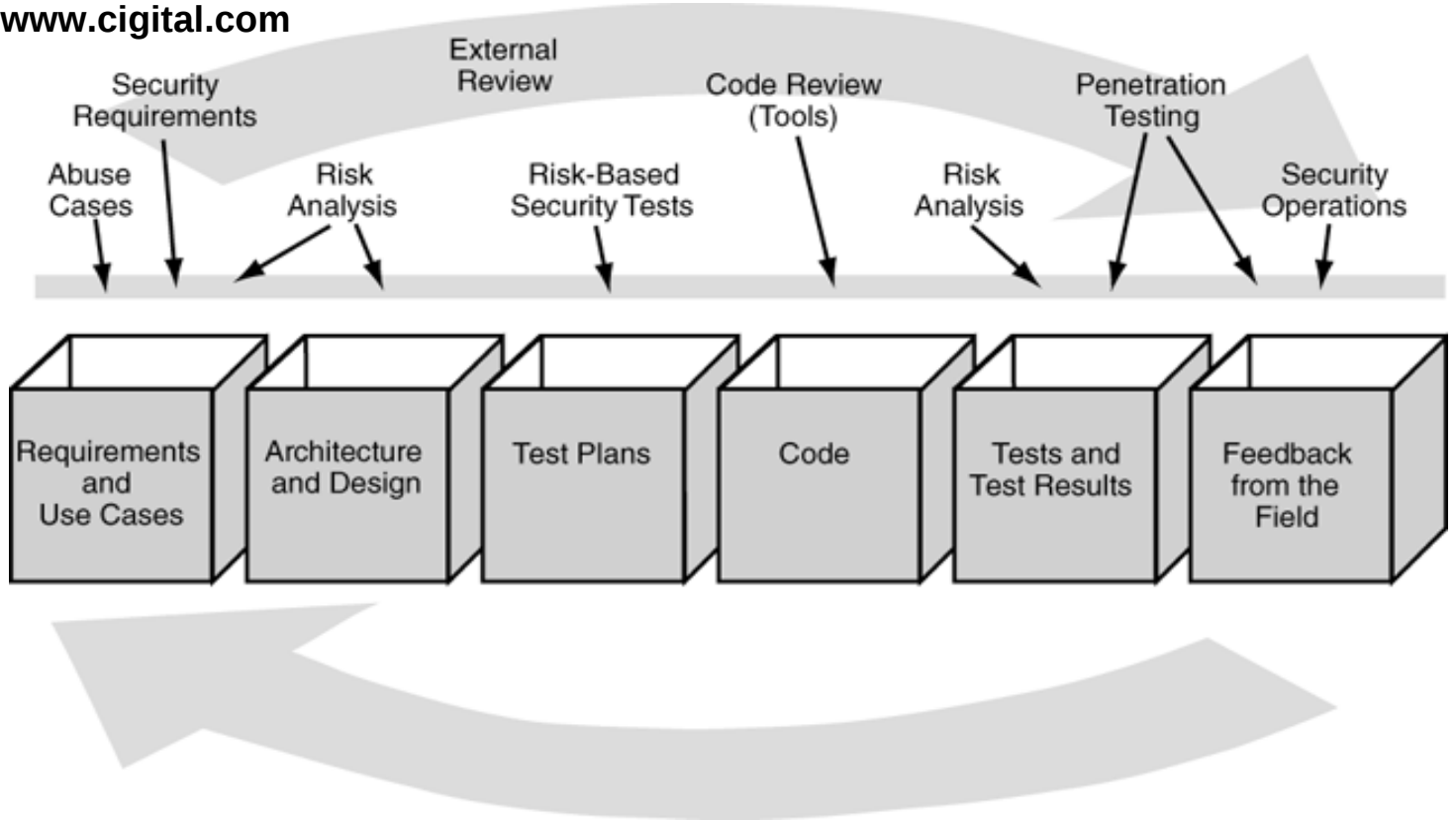
- **Auditoría de seguridad:**
  - Se busca que el sistemas cumpla con las normatividad del negocio y de la industria.

Seguridad y SDLC

Puntos de contacto (Touchpoints)

Cigital-Gary McGraw

http://www.cigital.com



Seguridad y SDLC

Puntos de contacto de Cigital

- **Dirigido por artefactos.**
- **Los puntos de contacto representan revisiones al proceso que se pueden efectuar en cada artefacto de desarrollo.**
- **Permite que el esfuerzo de seguridad se adapte a cualquier metodología SDLC.**
- **El principio rector es no alterar el proceso de desarrollo, sino integrarse profundamente con éste.**

Seguridad y SDLC

Puntos de contacto de Cigital

Punto de contacto 1: Casos de abuso

- Los casos de uso formalizan el comportamiento normal (y asumen uso correcto).
- Los casos de abuso o mal uso permiten:
  - Prepararse para comportamiento anormal (ataque)
  - Descubrir casos excepcionales
  - Aprovechar el hecho de que los diseñadores de saben más acerca del sistema que los posibles atacantes
  - Documentar explícitamente el comportamiento del software en situación de uso ilegítimo

Punto de contacto 2: Requerimientos de seguridad

- Algunos requerimientos de seguridad son evidentes
  - Los datos personales sensibles deben estar protegidos criptográficamente
  - Autenticación robusta de los usuarios
  - Cumplir con requerimientos regulatorios
- Los requerimientos deben estar claramente especificados
  - Control de sesiones persistentes y únicas
  - (“No permitir desbordamiento de buffer” no llega a ser un requerimiento)
- La especificación de requerimientos debe ser total
  - Tener en cuenta que un atacante necesita explotar solamente una vulnerabilidad

Seguridad y SDLC

Puntos de contacto de Cigital

Punto de contacto 3:
Análisis de riesgos de arquitectura

- Construir modelo simple del diseño
- Uso pruebas de hipótesis para clasificar los riesgos
  - Modelado de amenazas
  - Patrones de ataque
- Establecer rango de importancia de riesgos
- Vincular con el contexto de negocio
- Sugerir correcciones
- Repetir

- Punto de contacto 4:
  Pruebas de seguridad basadas en riesgos

- Pruebas de la funcionalidad de seguridad
  - Cubrir los requerimientos no funcionales
- Pruebas basada en riesgo
- Usar resultados de análisis de riesgos de arquitectura para conducir pruebas basadas en escenarios
- Concentrarse en lo que "no se puede hacer"
  - Pensar como un atacante

Seguridad y SDLC

Puntos de contacto de Cigital

Punto de contacto 5: Revisión del código

- La revisión de código necesariamente es maliciosa
- Las mejores prácticas de codificación hacen el trabajo más fácil
- Las herramientas automatizadas ayudan a detectar errores triviales
- Los errores de implementación sí importan
  - Ej.: los desbordamientos del búfer pueden ser descubiertos con análisis estático.
- La trazabilidad desde el punto vulnerable hacia los datos de entrada es crítica

Punto de contacto 6: Pruebas de penetración

- El software debe estar en su ambiente operativo
- Se observa como funciona el sistema completo en la práctica
  - -Interacción con los mecanismos de seguridad de la red
  - -Firewalls
  - -Uso de criptografía
  - Etc.
- Deben estar dirigidas por los riesgos no cubiertos a lo largo del ciclo de vida
- ¡No es una solución mágica!

Seguridad y SDLC

Puntos de contacto de Cigital

Punto de contacto 7:
Operaciones de seguridad

- Aprovechar el hecho de que los responsables de seguridad de la red deben tener mayor conocimiento de ataques reales
- Involucrar a las personas con conocimientos de seguridad en tantas actividades de “punto de contacto” como sea posible
- Configurar el ambiente operativo a las necesidades específicas de la aplicación:
  - Las configuraciones por omisión de los elementos de infraestructura normalmente no son seguras

Bibliografía

- Owasp. (http://www.opensamm.org) Software Assurance Maturity Model. A guide to building security into software development. Version 1.0.
- Talukder y Chaitanya, Architecting secure software systems, Ed. CRC Press, Auerbach Publications, 2009.
- Biskup, Security in Computing Systems. Challenges, Approaches and Solutions, Ed. Springer-Verlag, 2009.
- Allen el al, Software Security Engineering: A Guide for Project Managers, Ed. Addison Wesley Professional, 2008.
- Berg, High-Assurance Design: Architecting Secure and Reliable Enterprise Applications. Ed. Addison Wesley Professional, 2005.