Célula Académica UABC-Live .net

Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

http://uabc-live-net.spaces.live.com/

Sesión No. 5

Seguridad

Expositores:

Mario Ceceña Acosta **(lemariee@hotmail.com)**

Daniel Seara

Fecha: 3 de Marzo de 2007

Proceso de desarrollo

- Definir que hace la aplicación y como se usa
  - Los usuarios ven páginas con catálogos
  - Realizan búsquedas del mismo
  - Agregan ítems al carrito
  - Cierran la operación
- Diagrama de la aplicación
  - Mostrando
    - sub sistemas
    - Flujo de datos
    - Listando afirmaciones

Diseñando aplicaciones seguras

- Refinar el diagrama de la arquitectura
  - Mostrar mecanismos de autenticación
  - Mostrar mecanismos de autorización
  - Mostrar tecnologías (ej. DPAPI)
  - Diagrama de límites de confianza
  - Identificar puntos de entrada
- Empezar a pensar como el agresor
  - ¿Donde están mis vulnerabilidades?
  - ¿Qué puedo hacer para resolverlas?

Amenazas en una aplicación

Amenaza	Ejemplos
Inyección de SQL	Incluyendo comandos de SQL en el texto ingresado
Script entre sitios	Usando script del lado del cliente malicioso
Modificación del ingreso	Cambiando valores de campos ocultos
Robo de claves	Usando un investigador de paquetes (*sniffer*) para obtener claves de acceso o cookies no encriptadas
Reemplazo de sesión	Uso de Cookie de sesión “robado”, para acceder a sitios
Reemplazo de identidad	Uso de Cookie de Autenticación por Formulario, para hacerse pasar por otro
Revelado de información	Mostrar al cliente el seguimiento de la pila, cuando sucede un error

http://msdn.microsoft.com/library/en-us/dnnetsec/html/THCMCh10.asp?frame=true#c10618429_004

Identificando Amenazas

- Método #1: Lista de Amenazas
  - Comenzar con una lista base
  - Identificar las que se aplican a la App.
- Método #2: STRIDE
  - Lista categorizada de tipos de Amenazas
  - Identificar Amenazas por tipo
- Opcionalmente dibujar árboles
  - Los nodos raíz representan las metas del atacante
  - Los árboles ayudan a identificar las condiciones de Amenaza

Modelando amenazas

1. Identificar elementos
2. Documentar Arquitectura
3. Descomponer la Aplicación
4. Identificar Amenazas
5. Documentar Amenazas
6. Cuantificar Amenazas

STRIDE

Spoofing

Se puede acceder con una identidad falsa?

Tampering

Se pueden modificar datos mientras fluyen por la Aplicación?

Repudiation

Si se intenta denegar, se puede probar que es un agresor?

Revelado de información

Se puede acceder a información reservada?

Denial of service

Es posible disminuir la disponibilidad de la Aplicación?

Elevation of privilege

Puede un atacante asumir roles de usuario privilegiado?

DREAD

D **Damage**
¿Cuales son las consecuencias?

R **Reproducibility**
¿Se puede reproducir bajo ciertas circunstancias?

E **Exploitability**
¿Cuan fácil es realizarla?

A **Affected users**
¿Cuantos usuarios pueden verse afectados?

D **Discoverability**
¿Es fácil de descubrir?

DREAD, Cont.

	Alto (3)	Medio (2)	Bajo (1)
Daño Potencial	El agresor puede obtener datos muy sensibles, dañar servidores etc.	Puede obtener datos sensibles, pero casi nada más	Puede acceder a datos poco importantes
Reproductibilidad	Siempre es posible	Sucede si se realiza en un corto tiempo	Raramente se puede hacer
Explotabilidad	puede hacerlo	Se deben tener ciertos conocimientos	Tal vez alguno
Usuarios afectados	La mayoría	Algunos	Pocos, si es que hay alguno
Encubrimiento	Fácil de ver	Más costoso de ver	Muy difícil de encontrar

DREAD, Ejemplo

Amenaza	D	R	E	A	D	Sum
Robo de cookie aut. (Robo de claves)	3	2	3	2	3	13
Robo de cookie aut. (XSS)	3	2	2	2	3	12

Alto (Impersonalización, robo de claves etc.)

Es fácil, pero sirven sólo hasta su expiración

Cualquiera usa un sniffer; XSS requiere mayor experiencia

Muchos pueden ser afectados, pero cada vez menos gente abre mensajes de desconocidos

Fácil de encontrar: basta con escribir <script> en un cuadro de texto y probar

Riesgos Priorizados

Ejemplos de ataques

- Inyección de SQL (SQL Injection)
  - Es común por el armado de sentencias SQL encadenando valores ingresados por el usuario
  - Puede evitarse simplemente con la buena práctica de usar Procedimientos Almacenados
  - ASP.Net a partir de 1.1, valida automáticamente que no se envíe código inválido

Inyección de SQL

Desarrollador 5 Estrellas

Ejemplos de ataques

- Denegación de servicio (Denial of Service)
  - Evita el acceso a servicios generalmente al crear consumos inválidos
    - Consumo inválido de recursos
    - Utilización desmedida de recursos publicados (UDP)
    - Consumo de ancho de banda
    - Etc.

Plataforma de protección

- El esquema Firewall
- La DMZ (Zona des militarizada)

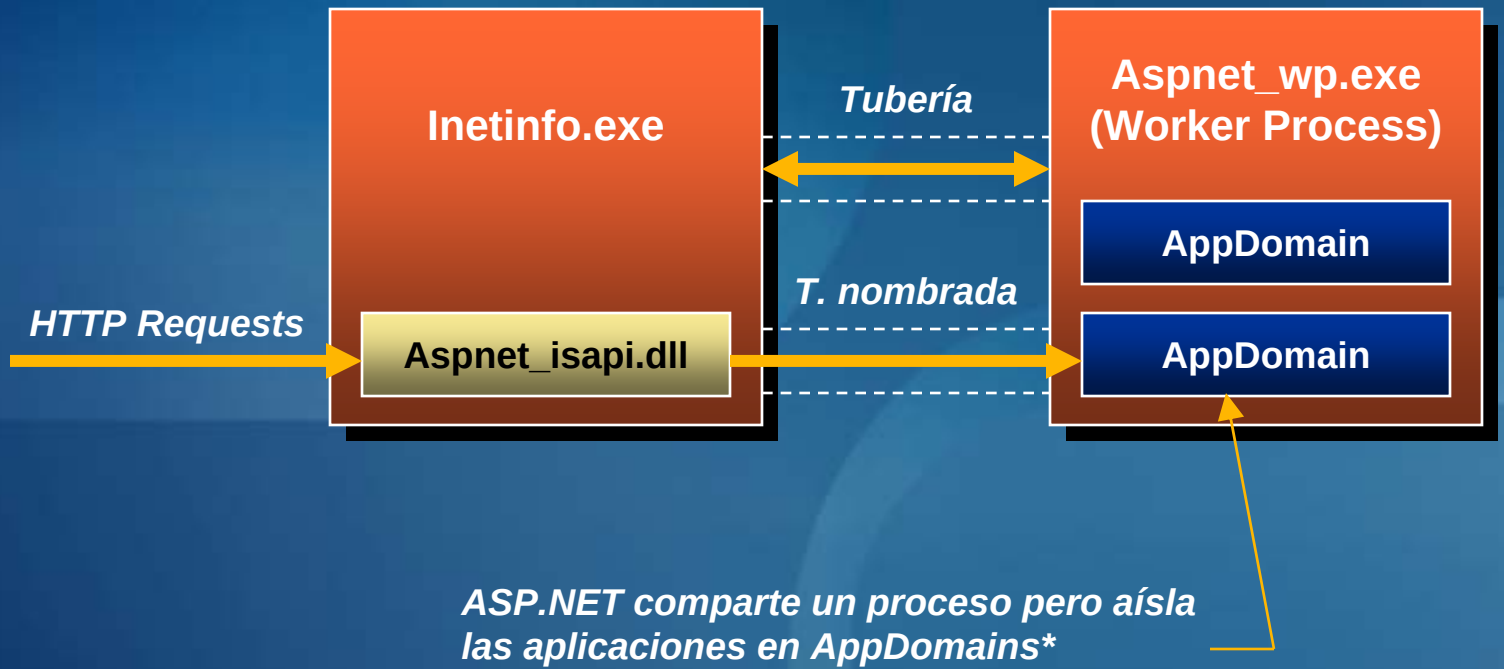
Autenticación y autorización

- Autenticación
  - Mecanismo para identificar un usuario
- Autorización
  - Mecanismo para permitir el acceso de un usuario específico, a un usuario específico
    - Por identidad
    - Por pertenencia

Mecanismos de Autenticación en IIS

	Basic	Digest	Windows Integrada		Certificado	Passport
			NTLM	Kerberos		
Necesita cuenta de Windows?	S	S	S	S	N	N
Soporta delegación?	S	S	N	S	S	S
Las credenciales van como texto plano?	S	N	N	N	N	N
Soporta navegadores no IE?	S	S	N	N	S	S
Pasa a través de firewalls?	S	S	N	N	S	S
Buena experiencia del usuario?	N	N	S	S	S	S

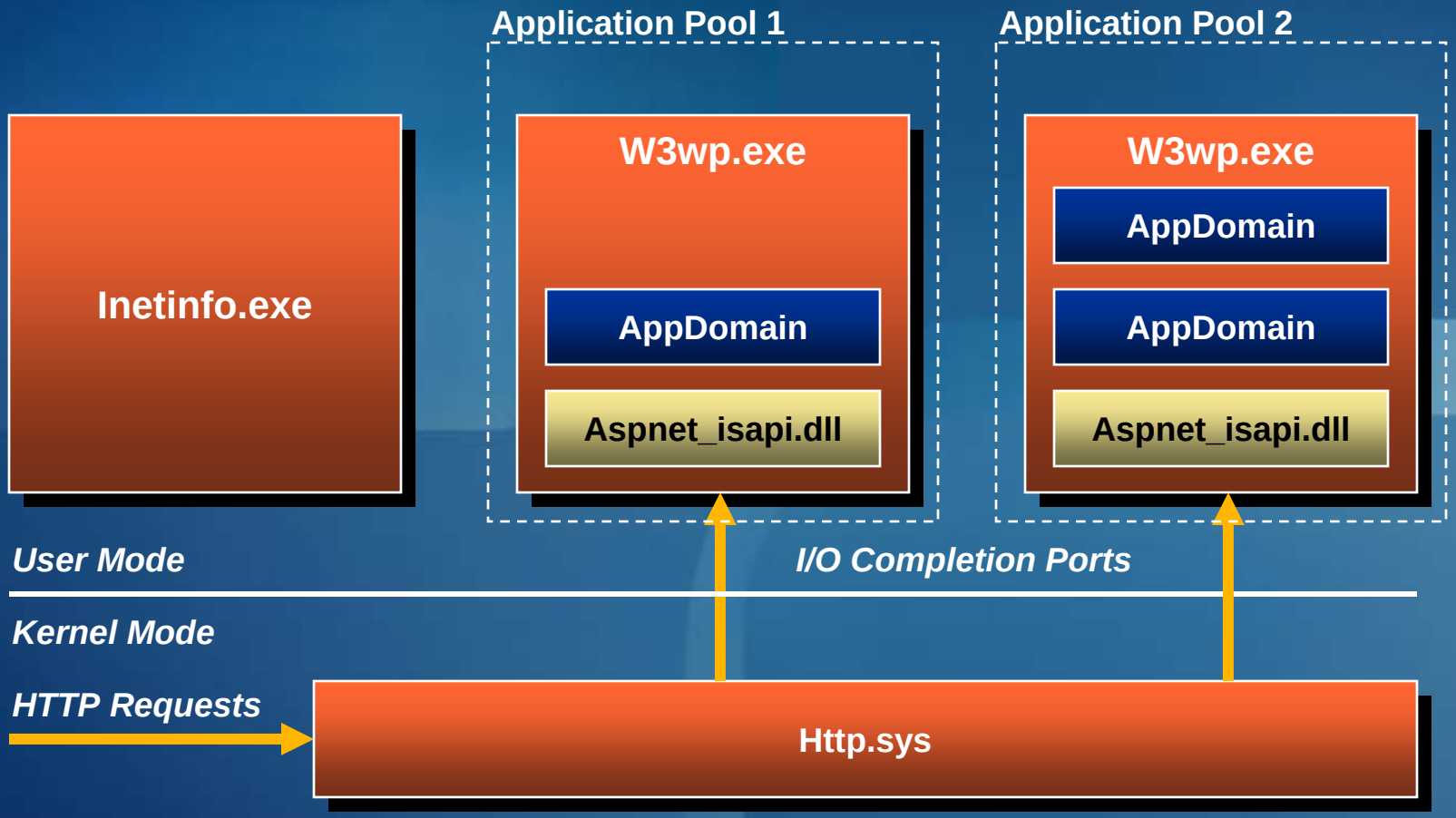
IIS 5 y ASP.Net



* ASP.NET también soporta jardines de servidores reservando un proceso por cada CPU

IIS 6 y ASP.Net

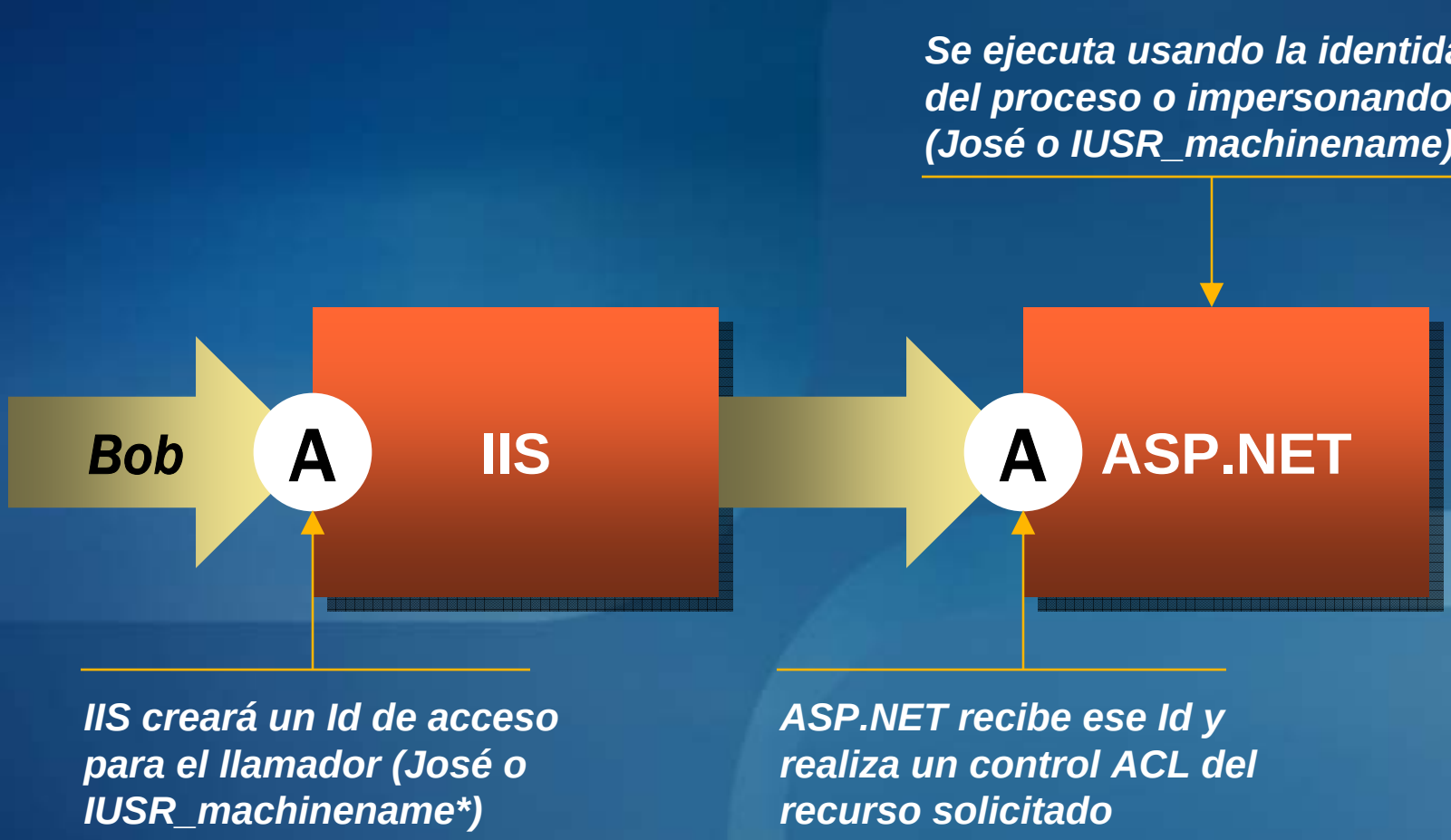
Application pooling: permite decidir como separar las aplicaciones entre los procesos de trabajo



Autenticación en ASP.Net

- Autenticación Windows
  - Utiliza usuarios existentes de Windows
  - Ideal para aplicaciones Intranet
- Passport
  - Conveniente para los usuarios (un solo inicio)
  - Deja el almacenamiento de credenciales en manos de otros
- Forms
  - Típicamente utilizada en aplicaciones de tipo carrito
  - Ideal para aplicaciones Internet

Autenticación Windows



* Normalmente, IIS usa IUSR_machinename para representar usuarios anónimos, pero puede ser cambiado

Por Formulario

Seguridad en Componentes

- Atributos de seguridad
  - Permite identificar el tipo de permisos que un tipo y/o sus métodos requieren para ser ejecutados
    - Demand
    - Permit
    - Assert
  - El componente debe tener un nombre firme (Strong name)
    - Sn
    - Versión

Seguridad en Base de datos

- Repasa los conceptos de seguridad en la presentación acerca de datos
  - Autenticación integrada
  - Cadena de conexión protegida
    - Encriptación
    - Ubicación fuera de la aplicación

Seguridad en Servicios

- Identificar adecuadamente al usuario que accede a un servicio
  - Integrada
  - Impersonalización
  - Mejoras para servicios Web (WSE)

Referencias

- How To: Perform a Security Code Review for Managed Code (.NET Framework 2.0)
  - http://msdn.microsoft.com/practices/default.aspx?pull=/library/en-us/dnpag2/html/PAGHT000027.asp
- Threat Modeling Web Applications
  - http://msdn.microsoft.com/practices/default.aspx?pull=/library/en-us/dnpag2/html/tmwa.asp
- Security Engineering Index
  - http://msdn.microsoft.com/practices/default.aspx?pull=/library/en-us/dnpag2/html/securityengindex.asp