



Programación segura de aplicaciones con tecnologías MS

Instituto para la Seguridad en Internet

Gonzalo Álvarez Marañón



Instituto para la
SEGURIDAD en INTERNET



Agenda

- ◆ Seguridad en el acceso a bases de datos
- ◆ Control de acceso
- ◆ SSL y certificados digitales
- ◆ Ataques de entrada de usuario
- ◆ Seguridad en SQL Server
- ◆ Arquitectura multicapa



Seguridad en el acceso a bases de datos desde ASP

- ◆ Cadena de conexión en cada página .asp
- ◆ Archivos de inclusión .inc, .asp, etc.
- ◆ Variables de aplicación en el global.asa
- ◆ Si se obtiene el código fuente, se obtiene la información de conexión a la BD
- ◆ ¿Es posible obtener el código fuente con un simple navegador?



Formas seguras de acceso

- ◆ Utilización de un componente COM con la cadena de conexión codificada
- ◆ La BD configurada para utilizar seguridad Windows, no estándar, usando conexión de confianza:

`Trusted_Connection`



Control de acceso

- ◆ Proteger la entrada a un web completo o sólo a parte:
 - *Autenticación*: identifica al usuario o a la máquina que trata de acceder a los recursos
 - *Autorización*: dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos



Autenticación básica

- ◆ El navegador presenta al usuario la ventana de autenticación
- ◆ El navegador intenta establecer una conexión con el servidor
- ◆ Si el servidor rechaza la información de autenticación, el navegador le presenta nuevamente la ventana al usuario
- ◆ Se establece la conexión de acceso al recurso protegido



Desventajas de este método

- ◆ Las contraseñas se transmiten en claro
- ◆ Los usuarios pueden revelar sus contraseñas a otras personas
- ◆ Los usuarios suelen perder u olvidar sus contraseñas
- ◆ Los usuarios no suelen elegir contraseñas especialmente robustas
- ◆ Los usuarios tienen la fea costumbre de apuntarlas en cualquier sitio





Autenticación de Windows integrada

- ◆ Nuevo nombre para la anterior “desafío/respuesta”
- ◆ Variante de la autenticación mediante resúmenes criptográficos, pero negociando NTLM o Kerberos
- ◆ El navegador tiene que demostrarle al servidor que conoce la clave por medio de un corto intercambio de datos



Autenticación de Windows integrada

◆ Ventajas

- No transmite las contraseñas en claro a través de Internet

◆ Desventajas

- Sólo está soportado por IE, versión 2.0 o posterior y servidores NT
- No funciona para conexiones con proxy



Autenticación a nivel de aplicación

- ◆ Tabla con los nombres de usuario y sus contraseñas
- ◆ `strSQL = "SELECT IdUsuario "_ &"FROM Users "_ &"WHERE login ='"&login&" AND password = '"&password&"'"`
- ◆ Si las credenciales son válidas, se recupera el ID del usuario y se le permite acceder al recurso/servicio



Limitaciones

- ◆ Ataques de SQL
- ◆ Contraseñas en claro en la BD
- ◆ Ataques de diccionario, ya que no existe bloqueo por intentos de login fallidos
- ◆ Credenciales sólo en la página de entrada: ¿y el resto?



SSL y certificados digitales

- ◆ SSL: protocolo para dotar de seguridad a las sesiones de navegación a través de Internet
- ◆ Creado en 1994 por Netscape Communications Corporation junto con su primera versión del Navigator
- ◆ Con SSL v3.0 alcanzó su madurez





Servicios de seguridad de SSL

- ◆ Confidencialidad: cifrado de datos
- ◆ Integridad de mensajes
- ◆ Autenticación de servidores
- ◆ Autenticación de cliente (opcional)



Servidores seguros

- ◆ SSL no es sinónimo de seguridad:
 - Agujeros en guiones CGI, ASP, ISAPI, etc.
 - Páginas mal diseñadas que permiten acceder a cuentas de otros usuarios
 - Ficheros confidenciales accesibles a través de web
 - Servidor con muchos agujeros de seguridad



Limitaciones comerciales de SSL

- ◆ SSL ofrece canal seguro, pero carece de proceso financiero
- ◆ SSL garantiza confidencialidad en tránsito, no en el servidor
- ◆ Proporciona falsa sensación de seguridad: cualquiera puede montar un sitio web seguro
- ◆ Debilidad criptográfica: todavía se ven aplicaciones con 40 bits de clave



Control de acceso por certificados

- ◆ Los clientes se identifican mediante la presentación de un certificado
- ◆ Contienen información como su nombre, empresa, departamento, dirección de correo, ciudad, país, etc.
- ◆ Permite sofisticados esquemas de control de acceso basándose en uno de estos atributos o en una combinación de varios o en el conjunto del certificado



Cómo funcionan

- ◆ El usuario instala el certificado en su navegador
- ◆ Sólo tiene que presentarlo para que se produzca la autenticación
- ◆ Pueden almacenarse localmente en el disco duro o en una tarjeta inteligente
- ◆ Están además protegidos por una contraseña



Correspondencias de certificados y cuentas

- ◆ Uno a uno: se puede hacer corresponder uno o varios certificados individuales a cada cuenta
- ◆ Muchos a uno: utilizan reglas de ajuste basándose en la información contenida en los distintos atributos de los certificados, aceptándose todos los certificados de cliente que verifiquen ciertas reglas



Certificados digitales

◆ Ventajas

- Permiten autenticarse en muchos servidores distintos
- Fáciles de escalar
- Permiten descentralizar la verificación de permisos de acceso

◆ Desventajas

- Quién emite los certificados
- Qué confianza proporciona un certificado
- TALÓN DE AQUILES: CRL
- Elevada complejidad





Ataques de entrada de usuario

- ◆ Inyección de SQL: consultas maliciosas
- ◆ Desbordamientos de búfer
- ◆ Cross-site Scripting e inyección de código
- ◆ Validación de entrada en cliente y servidor
- ◆ Campos ocultos
- ◆ Cabecera Referer
- ◆ ¿POST o GET?





Inyección de SQL

- ◆ Ejecutar consultas que la aplicación no está preparada para aceptar
- ◆ Causas:
 - No se valida la entrada
 - No se establecen permisos de acceso en los objetos de la BD
 - Se construyen las consultas directamente desde ASP



Desbordamientos de búfer

- ◆ Un búfer se desborda cuando se intenta meter en él más cosas de las que caben en el espacio que tenía reservado
- ◆ Consecuencias del desbordamiento:
 - El programa deja de funcionar
 - Posibilidad de ejecutar código arbitrario en su contexto de seguridad
- ◆ Todos los programas en todos los SO padecen este problema



Cross-site Scripting e inyección de código

- ◆ El atacante introduce código JavaScript arbitrario en un enlace, que se ejecutará en el navegador de la víctima bajo el contexto del servidor atacado
- ◆ El código puede insertarse en hiperenlaces enviados por correo o en páginas web
- ◆ O puede inyectarse en una página, como un libro de visitas, foros de discusión o comentarios sobre productos





Validación de entrada

- ◆ Es fundamental validar los datos que introduce el cliente
- ◆ ¿Basta con validar en el cliente o reproducir todas las validaciones en el servidor?
- ◆ Expresiones regulares: Objeto RegExp



Campos ocultos

- ◆ `<input type="hidden">`
- ◆ Son invisibles en la página web...
- ◆ ...¡pero no al editar el código fuente!
- ◆ La información sensible en campos ocultos puede ser fácilmente manipulada
- ◆ ¿Soluciones?



Referer

- ◆ El navegador entrega rutinariamente al servidor el URL de la página desde la que procede
- ◆ Suele utilizarse para validar que los formularios se envía desde la página esperada
- ◆ Se pueden manipular a mano





¿POST o GET?

- ◆ Historial del navegador
 - Al rellenar un formulario con GET, todos los valores introducidos quedan en el Historial
- ◆ Registros del servidor
 - También quedan en los logs de IIS
- ◆ Cabecera HTTP_REFERER
 - También se transmiten en la cabecera Referer



Cifrado de scripts

- ◆ El código en el cliente son visibles por cualquiera
- ◆ El código de las páginas ASP también puede acabar en manos de un atacante
- ◆ Posibilidad de codificar su código para IE 5.0+.

`screnc entrada salida`

- ◆ Se decodifica con `scrdec` o similares

Seguridad en SQL Server



SO Windows 2000

SQL Server

Base de datos

Objetos de BD





Tipos de autenticación

- ◆ Autenticación Windows 2000
 - Usuarios con cuentas de Windows
 - El SO se encarga de la autenticación (SIDs)
 - Conexiones de confianza
- ◆ Autenticación SQL Server
 - Usuario propio de SQL Server, no del SO
 - SQL Server gestiona la autenticación
 - Conexiones no fiables
 - Limitaciones de las contraseñas



Identificación ante la BD

- ◆ Necesario crear usuarios de BD y asignar inicios de sesión a esos usuarios
- ◆ Conviene definir usuarios distintos para distintas BD (compartimentación)
- ◆ Los inicios de sesión pueden ser usuarios o grupos de W2K o de SQL Server
- ◆ Además, cada usuario debe tener permiso sobre los objetos de la BD



Procedimientos almacenados

◆ Ventajas:

- Mejoran el rendimiento porque el código está compilado
- Pueden encadenar multitud de consultas
- Facilitan el mantenimiento
- Aumentan la seguridad
 - Control de ejecución de sp y no sobre las tablas

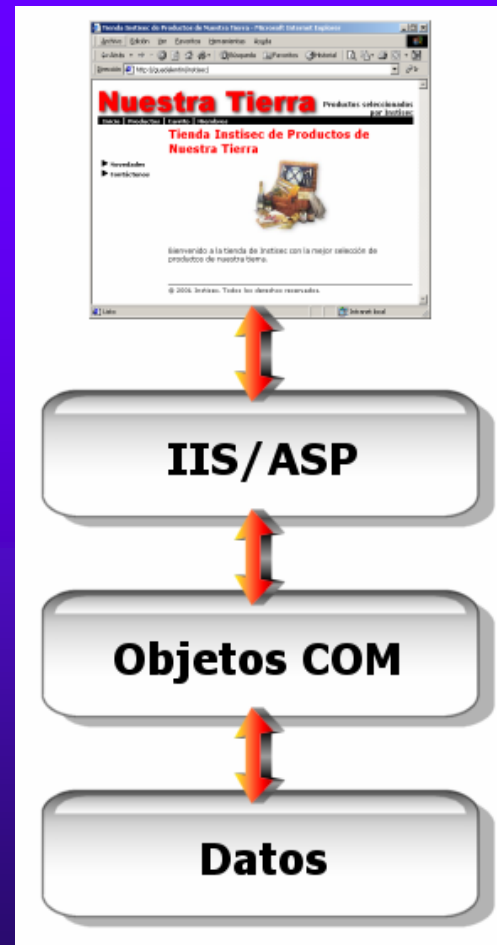




Puertas traseras

- ◆ Escalada de privilegios explotando xp_cmdshell: creación de un usuario y adición al grupo Administradores
- ◆ sqlpoke: búsqueda de servidores con contraseña de sa en blanco
- ◆ sqlping: información sobre servidores
- ◆ sqlexec: shell en el servidor
- ◆ sqlbf: ataque de fuerza bruta sobre usuarios del servidor

Arquitectura multicapa





Ventajas

- ◆ Escalabilidad
- ◆ Seguridad
- ◆ Separación en capas lógicas:
 - Servicios de usuario: presentación
 - Servicios de negocio: lógica
 - Servicios de datos



Seguridad en componentes COM+

- ◆ Modelos de seguridad:
 - Declarativo: la seguridad se configura a través del explorador de servicios de componentes
 - Programático: los controles de seguridad se realizan dentro del propio código del componente



Configuración del control de acceso

- ◆ Grano grueso: se controla el acceso a toda la aplicación COM+
- ◆ Grano fino: se controla el acceso a:
 - Componentes individuales de la aplicación
 - Interfaces individuales de un componente
 - Métodos individuales de una interfaz