Programación segura con .NET

Mario Guzmán
Desarrollador
Mario.Guzman@gmail.com

OWASP

Aspectos a ser Tratados

- Como guardar de manera segura Passwords de Usuarios
- Limpiar la entrada de datos

Manejar Passwords

Ataques
Como Guardar Passwords

OWASP

Mecanismos de Ataque

- Fuerza Bruta
- Ataques de Diccionario
- Brechas de Seguridad en el Servidor de Bases de Datos

Fuerza Bruta y Ataques de Diccionario

- Fuerza Bruta: consiste en probar con todas las combinaciones posibles de letras, números etc.
- Ataque de Diccionario: prueba con palabras de diccionario, fechas, etc.
- Ambos necesitan un programa que mande passwords a una pagina web y verifique su resultado.

Ayuda de GPUs

- Herramienta usada ighashgpu
- NTLM
- Se compara el resultado del hash del password
- *ATI 5770*
  - 9.8 millones de passwords/sec

GPU Fuerza Bruta

- 5 caracteres: 4 segundos
- 6 caracteres: 17 segundos
- 7 caracteres: 17 minutos 30 segundos.
- 8 caracteres: 18 horas y 30 minutos
- 9 caracteres: 48 dias.

Defensa

- Pedir Passwords de 8-9 caracteres.
- Poner Timers: Dejar solo probar 10 logins seguidos
- Idealmente pedir Passphrases
- Tener un diccionario de las palabras mas usadas y no dejar que le gente use estos passwords: Mama, Papa, etc.
- Pedir passwords alfa numéricos.

BRECHAS DE SEGURIDAD EN UNA DB

Riesgos

- Debilidad en el código que permita tener acceso a las tablas de usuarios.
- SQL Injection
- DB Admins maliciosos.

Que queremos lograr

- Que aunque la base de datos caiga en manos maliciosas los passwords de los usuarios no sean comprometidos
  - Usuarios usan el mismo password para varios sitios

Hash

- No guardar como “Clear Text” un password
- Usar un algoritmo de Hash
- Hash es un algoritmo de una sola via: cuando se aplica un hash no se puede recuperar el password otra vez.
- Al momento de login, se le pide al usuario el password, y se aplica el hash al mismo y por ultimo se comparan los hashes, nunca los passwords.

Hash

- Ejemplo en código

Hash + Salt

- En caso que el atacante tenga acceso a la base de datos entera, puede todavia tratar de adivinar los hashes por metodos de fuerza bruta

Hash + Salt

- Salt son bytes/string generado al random
- Se concatena/xor al password original
- Se logra tener un password bastante mas largo
- Tiene la desventaja que se tiene que guardar idealmente en otra base de datos

Hash + Salt

- Ejemplo

Hash + Salt + Iterations

- Se toma el resultado del hash + salt y se vuelve a pasar por la función de hash + salt n veces.
- El número de iteraciones debe de ser random y se recomienda que sea de mas de 1000 veces.
- Tiene que ser guardado junto al salt

Hash + Salt + Iterations

- Ejemplo en código

Limpiar Entradas a DB

Riesgos
SqlParameters
LINQ to SQL

OWASP

Riesgos

- SQL Injection
- Cross Site Scripting (XSS)
- Aunque verifiquemos los strings en el 99.99% de nuestra aplicación solo es necesario que falle una vez para ser vulnerables

SQL Injection

- Pasar comandos de SQL como parte de los valores
- Ejemplo en código para
  - SQL Parameters
  - Linq to SQL

Cross Site Scripting

- Pasar valores con comandos de Javascript/html
- Ejemplo: <b onmouseover=alert('Wufff!')>click me!

Otros Métodos

- Nhibernate
- Fluent Nhibernate
- Object Relational Mapping

Preguntas y/o comentarios

OWASP