Laboratorio III

SQL Injection

Introducción

Al trabajar en diversas arquitecturas siempre existe la posibilidad de dejar abscesos a nuestros sistemas de software, esto puede darse por inexperiencia, accidente y a veces a propósito para generar “atajos”.

Al seguir “Bad Practices” existe la gran posibilidad de sacrificar la seguridad de un sistema.

Uno de los peores es la Inyección de SQL, la cual permite el ingreso a consultas y métodos de una base de datos desde una capa de usuario, debido a una mala implementación o fallos en separación de capas y responsabilidades.

¿Qué es?

Consiste en el ataque por medio de la inserción de un Query de SQL en un Input dentro de un sistema computacional. Su propósito va desde el robo de datos hasta la eliminación de los mismos.

¿Como se puede evitar?

Existen diversas formas de evitar este tipo de ataques. La principal es la validación de campos de manera efectiva para evitar el ingreso de datos no autorizados en los formularios del sistema.

Además, nunca es remendado construir secuencias de SQL para su posterior envió a la base de datos, la parametrización consiste en el acceso de acceso y validación de parámetros en los diferentes Clientes de consulta SQL.

Ejemplo

```
Imports System.Data.SqlClient

Module MainModule
    Sub Main()
        Dim ced As String
        Do
            Console.WriteLine("Inserte Su Numero De Cedula")
            ced = Console.ReadLine()
            If (ced <> "-1") Then
                SQLRequest(ced)
            End If
            Console.Clear()
        Loop While ced <> "-1"
    End Sub
    Sub SQLRequest(ByVal Optional cedula = "")
        Dim DB As SqlConnection = New SqlConnection("Data Source=.;Initial
Catalog=SQLInjection;Integrated Security=True")
        DB.Open()
        ' Here You Should Set this as parameters otherwise you can send "116810122
or Nombre != ''"
        Dim Command As SqlCommand = New SqlCommand("SELECT * FROM USUARIO WHERE ID =
" + cedula, DB)
        Command.Prepare()
        Dim Reader As SqlDataReader = Command.ExecuteReader()
        If Reader.HasRows Then
            While (Reader.Read())
                Console.WriteLine("ID:" + Reader("ID"))
                Console.WriteLine("Nombre:" + Reader("Nombre"))
                Console.WriteLine("Apellido:" + Reader("Apellido"))
                Console.WriteLine("Direccion:" + Reader("Direccion"))
                Console.WriteLine("Correo:" + Reader("Correo"))
            End While
        End If
        Console.ReadKey()
    End Sub
End Module
```

Bibliografía

SQL Injection. (n.d.). Retrieved February 10, 2018, from https://www.owasp.org/index.php/SQL_Injection