Caso Privado X

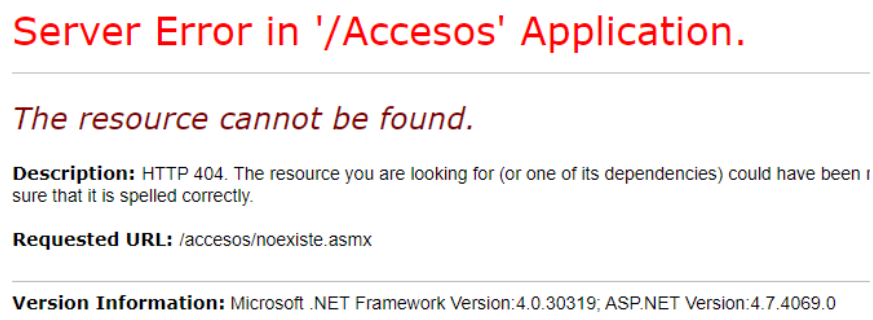
Servicio Web

URL: https://nombreservidor/accesos/webservice.asmx

Reconocimiento:

Por la extension ASMX del archivo, se reconoce que esta desarrollado en .NET.

Tratando de acceder a un archivo inexistente NOEXISTE.ASMX, se reconoce la versión exacta del Framework, y a que además posee la pantalla por defecto de error conocida como YSOD (Yellow Screen of Death).



En la pagina principal se logran enumerar los métodos que posee el WS, y que además contiene el namespace por defecto.

Se puede acceder al documento WSDL a través del link “Service Description”, donde se reconoce el acceso al WS por los métodos GET, POST, SOAP y SOAP12.

Método Auth: solamente se obtiene respuesta True/False, tratando de provocar distintos errores. No se encontraron vulns.

Método Cipher: se descubre que no hay manejo de errores. Se descubre el funcionamiento: texto es el valor que se quiere encriptar/desencriptar, key es la clave que se usa para encriptar/desencriptar, mode 1 encripta 0 desencripta. No se encontraron vulns.

Método EnviaCorreo: posee un manejo básico de errores. Permite el envio de mails al exterior. No se encontraron vulns.

Método Token: se descubre que no hay manejo de errores. Se descubren rutas donde busca archivo según el parámetro ingresado.

```
System.IO.FileNotFoundException: Could not find file &#39;\\                    \CIPHER\.CIPHER&#39;.
   at System.IO.__Error.WinIOError(Int32 errorCode, String maybeFullPath)
   at System.IO.FileStream.Init(String path, FileMode mode, FileAccess access, Int32 rights, Boolean useRigh
FileOptions options, SECURITY_ATTRIBUTES secAttrs, String msgPath, Boolean bFromProxy, Boolean useLongPath,
   at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 buffe
msgPath, Boolean bFromProxy, Boolean useLongPath, Boolean checkHost)
   at System.IO.StreamReader..ctor(String path, Encoding encoding, Boolean detectEncodingFromByteOrderMarks,
   at System.IO.File.InternalReadAllText(String path, Encoding encoding, Boolean checkHost)
   at System.IO.File.ReadAllText(String path)
   at WebService.Token(Int32 Test, String ID)
```

Se descubre una carpeta compartida en el mismo servidor (CIPHER) a la cual se puede acceder.

Se prueban los nombres de archivos.

Se obtienen textos en base64.

```
<string xmlns="http://tempuri.org/">fx7RnE8Edmhf1oD0VGYaES3+0R5C+eeyWRhl2H4A9kJzxgVkH0n9v[redacted]LEpBuwjOXzK0FAOMm3MTWNYQ==</string>
```

Probamos esta info en el método Cipher:

Con la KEY tal cual se obtuvo, no funciona, genera error. Observando el formato parece base64, se decodifica.

Con la nueva KEY, que es legible, se logra desencriptar el texto.

Es un string de conexión a SQL Server, en principio.

```
<string xmlns="http://tempuri.org/">                                |sqluser|                </string>
```

Utilizando SQL Management Studio, se logra conectar con las credenciales, logrando acceder a todas las bases de datos pudiendo leer y modificar los datos de las tablas.