

Caso Privado X

Servicio Web

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

Reconocimiento:

Por la extensión ASMX del archivo, se reconoce que está 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).

Server Error in '/Accesos' Application.

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been in a file that is no longer available, has been moved, or has been renamed. Please make sure that it is spelled correctly.

Requested URL: /accesos/noexiste.asmx

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.7.4069.0

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

WebService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Auth](#)
- [Cipher](#)
- [EnviaCorreo](#)
- [Hola](#)
- [Token](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

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.

```
<?xml version='1.0' encoding='utf-8'>
<wsdl:service name="WebService">
  <wsdl:port name="WebServiceSoap" binding="tns:WebServiceSoap">
    <soap:address location="https://nombreservidor/accesos/webservice.asmx"/>
  </wsdl:port>
  <wsdl:port name="WebServiceSoap12" binding="tns:WebServiceSoap12">
    <soap12:address location="https://nombreservidor/accesos/webservice.asmx"/>
  </wsdl:port>
  <wsdl:port name="WebServiceHttpGet" binding="tns:WebServiceHttpGet">
    <http:address location="https://nombreservidor/accesos/webservice.asmx"/>
  </wsdl:port>
  <wsdl:port name="WebServiceHttpPost" binding="tns:WebServiceHttpPost">
    <http:address location="https://nombreservidor/accesos/webservice.asmx"/>
  </wsdl:port>
</wsdl:service>
```

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 envío 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 useRights, FileOptions options, SECURITY_ATTRIBUTES secAttrs, String msgPath, Boolean bFromProxy, Boolean useLongPath, Boolean useDefaultPermissions, String msgPath, Boolean bFromProxy, Boolean useLongPath, Boolean checkHost)
  at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, String fileName, Boolean useLongPath, Boolean checkHost)
  at System.IO.StreamReader..ctor(String path, Encoding encoding, Boolean detectEncodingFromByteOrderMarks, Boolean checkHost)
  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+eeyWRh12H4A9kJzXgVkH0n9v...LpBuwjOXzK0FA0Mm3MTWNYQ==</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.