POSIBLES ATAQUES QUE PUEDE RECIBIR UN API SOAP

XML Injection (Inyección de XML)

Definición:

Consiste en que el atacante introduce código malintencionado en las etiquetas del XML como una inyección SQL

Como puede perjudicarnos:

Si un usuario introduce código SQL y nosotros manejamos las consultas a la base de datos concatenando el valor directamente el código SQL malintencionado del atacante formaría parte de nuestra consulta y podría llegar a obtener permisos en el sistema.

Daños al sistema como la perdida de datos por eliminación

Robo o filtración de información

Como podemos mitigar estos riesgos:

Podríamos hacer validaciones y sanitización de datos para eliminar caracteres como: ‘ “ - < > ;

Pero la mejor manera de prevenir esto es no construir la consulta concatenando datos sino utilizando parámetros para que el motor de la base de datos los tome como cosas separadas, y todo como una sola unidad.

XXE (XML External Entity)

Definición:

Consiste en que el atacante hace una referencia a una entidad externa, como un archivo dentro del sistema donde se ejecuta el programa, o relacionando un archivo externo en la web por medio de una URL

Como puede perjudicarnos:

El atacante puede llegar a leer archivos internos del servidor en donde se aloja nuestra API SOAP como los archivos de configuración.

Además de eso podría cargar un archivo externo por medio de una URL el cual puede ser muy pesado provocando la denegación del servicio DoS o malicioso.

Como podemos mitigar estos riesgos:

En .NET se puede configurar el lector de XML **XMLReader** y deshabilitar la ejecución de archivos o redirecciones de URLs

DoS por XML Bomb

Definición:

Es un XML que contiene definiciones de entidades que se auto referencian una un número de veces a la anterior sucesivamente haciendo que el número de caracteres crezca exponencialmente hasta dejar el servidor sin memoria.

Como puede perjudicarnos:

Debido a que el número de caracteres pueden crecer de manera exponencial, dependiendo de que tantas veces que la entidad se referencie puede llegar a llenar la memoria del servidor y provocando una caída del mismo, perjudicando fuertemente la API.

Como podemos mitigar estos riesgos:

Deshabilitando la expansión de **DTDs** creando una configuración **XmlReaderSettings**,

Limitando el tamaño del **XML** desde **readerSettings** para evitar que el **XML** crezca sin límites.

Usar parsers modernos para leer estos **XML**; En vez de usar **XmlDocument**, usar **XmlReader** con una configuración segura.

Man-in-the-Middle (MitM)

Definición:

Este ataque consiste en que el atacante intercepta la comunicación entre dos partes, cliente y servidor para leer o incluso modificar el contenido de sus mensajes, solicitudes o respuestas.

Cómo puede perjudicarnos:

Si el atacante tiene acceso al contenido de nuestros mensajes, como usuarios y contraseñas en una solicitud de login, por ejemplo, podría suplantar la identidad de nuestro usuario o reenviar un mensaje con un token valido.

También podría modificar el contenido de nuestra petición o la respuesta que nos llega desde el servidor, e igualmente de forma inversa. Y todo eso sin que nosotros nos demos apenas cuenta.

Cómo podemos mitigar estos riesgos:

Utilizando Https en vez de Http, y forzar la conexión por Https, para mantener el trafico cifrado para que si alguien pudiere de alguna forma interceptar algún mensaje el contenido de la comunicación no sea útil ni legible.

Firmar el menaje SOAP con WS-Security, para firmarlo digitalmente así evitando que alguien modifique el contenido, ya que las firmas no coincidirán si lo hacen, y luego validar la firma también desde el cliente.

No exponer datos innecesarios o sensibles para transportar la menor cantidad posible de datos sensibles.