SECUINFOR

Centro de Operaciones
de Ciberseguridad

#SiempreAlerta

REVISANDO TÉCNICAMENTE CADA UNO DE LOS RIESGOS PRESENTES EN LAS APLICACIONES WEB

A1 INYECCIÓN DE CÓDIGO

Autor: Christian Mendoza Bonzo.

Objetivo: Describirles a los desarrolladores y gerentes de proyectos en desarrollo en qué consiste el riesgo de *Inyección de Código*, alternativas para detectarlo y los mecanismos de protección,

Fecha: Septiembre 20, 2016.

Continuando con la descripción de los riesgos en aplicaciones, ahora desarrollaremos la descripción de cada uno de los riesgos, cómo identificarlas y qué acciones podemos tomar.

A1- Inyección de Código.-

Recordemos que esta vulnerabilidad ocurre cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta, el resultado es engañar a este intérprete para que ejecute dichos comandos y así acceder a datos no autorizados.

Revisemos paso a paso cómo se puede generar este tipo de amenazas antes que se transformen en un riesgo.

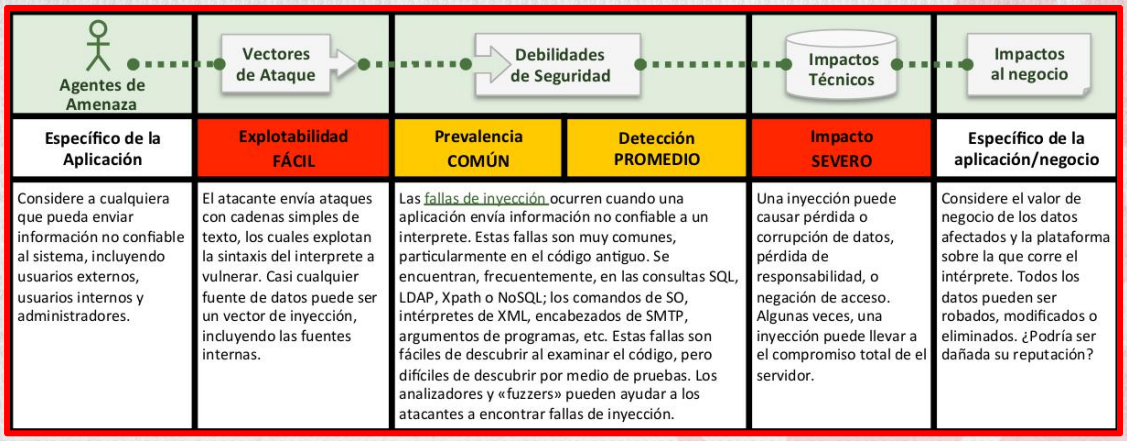

Fig 1. Inyección de Código. Copyright por: Owasp. Re-impreso con Permiso

¿Cómo conocer si soy vulnerable?

Existen varias formas de conocer, una de ellas es verificando que en todo uso de intérpretes se separa la información sensible o no confiable de la ejecución de una consulta o comando; para efectuar una llamada a una base de datos SQL desde una aplicativo debemos de usar variables parametrizadas en todas las sentencias preparadas y procedimientos almacenados o funciones, evitando las consultas dinámicas.

Otro mecanismo es verificar el código con herramientas de análisis de código, ya que estas pueden ayudar al analista de seguridad a ver cómo se utilizan los intérpretes y permiten seguir el flujo de datos a través de la aplicación.

Por último, existe el análisis dinámico automatizado, el cual puede incluir pruebas estáticas y dinámicas, esto dependerá de cada herramienta, sin embargo debe de ser considerado dado que puede brindar mayor granularidad en la detección de brechas de seguridad así como de vulnerabilidades, y

evidenciar si es necesario enriquecer el manejo de errores para dificultar que las inyecciones sean complicadas de descubrir.

¿Cómo prevenirlo?

Para evitar una inyección de código se requiere mantener los datos sensibles y no confiables separados de la ejecución de comandos, consultas y funciones.

Existen buenas prácticas que se pueden tomar, como se detalla a continuación:

1. La primera opción es usar una API segura, la cual evite el uso de intérpretes por completo o provea una interface parametrizada.

2. Si una API parametrizada no está disponible, debe codificar cuidadosamente los caracteres especiales y de manejo de errores, usando la sintaxis de escape específica del intérprete.

3. Es importante efectuar la validación de entradas positivas, recuerde que no es un mecanismo de defensa integral dado que en ocasiones las aplicaciones requieren caracteres especiales en sus entradas.

4. También existen dispositivos de seguridad que pueden brindar un mecanismo de protección como los denominados Web Application Firewalls (WAF); pero estos dispositivos deben ser administrados y actualizados constantemente por la aparición de nuevas técnicas de inyección.

Si desea conocer cómo podemos ayudarlo a proteger su información confidencial contáctenos a:

Email: info@secuinfor.com
PBX: +593-4-3726620
Website: http://www.secuinfor.com