Lista de Chequeo: Manejo de entrada de datos maliciosos. Tabla 5

- Los campos de entrada en la aplicación están siendo validados adecuadamente y cumplen el propósito que se prevé para él.

- No se confía y se trata como información maliciosa a algún dato externo de otras entidades o de algún cliente.

Descripción	Nivel		
	1	2	3
Verificar que el entorno de ejecución no es susceptible a desbordamientos de búfer o que los controles de seguridad previenen desbordamientos de búfer.	X	X	X
Verificar que las fallas de validación de entradas de datos del lado del servidor sean rechazadas y registradas.	X	X	X
Verificar que se aplican las rutinas de validación de entradas de datos del lado del servidor.	X	X	X
Verificar que un único control de validación de entrada es utilizado por la aplicación para cada tipo de datos que es aceptado.			X
Verificar que todas las consultas de SQL, HQL, OSQL, NOSQL y procedimientos almacenados, llamadas de procedimientos almacenados están protegidos por la uso de declaraciones preparadas o parametrización de consultas y por lo tanto no sean susceptibles a la inyección de SQL .	X	X	X
Verificar que la aplicación no es susceptible a la inyección LDAP, o que los controles de seguridad previenen inyección LDAP.	X	X	X
Verificar que la aplicación no es susceptible a la inyección de comandos del sistema operativo o que los controles de seguridad previenen la inyección de comandos del sistema operativo.	X	X	X
Verificar que la aplicación no es susceptible a la inclusión de archivo remoto (RFI) o inclusión de archivo Local (LFI) cuando el contenido es utilizado como una ruta a un archivo.	X	X	X
Verificar que la aplicación no es susceptible a ataques comunes de XML, como manipulación de consultas XPath, ataques de entidad externa XML y ataques de inyección XML.	X	X	X
Asegurar que todas las variables string utilizadas dentro de HTML u otro lenguaje web interpretado en cliente se encuentra apropiadamente codificado manualmente o se utiliza plantillas que automáticamente codifican contextualmente para asegurar que la aplicación no sea susceptible a ataques **DOM Cross-Site Scripting (XSS)**.	X	X	X
Si el framework de la aplicación permite asignación automática de parámetros en masa (también llamada enlace automático de variables o variable binding) desde la petición entrante a un modelo, verificar que campos sensibles de seguridad como "***accountBalance***", "***role***" o "***password***" están protegidos de enlaces automáticos maliciosos.		X	X

Verificar que la aplicación contenga defensas contra los ataques de contaminación de parámetros HTTP (agregar parámetros a la URL), particularmente si el framework de la aplicación no hace distinción sobre el origen de los parámetros de la petición (GET, POST, cookies, cabeceras, ambiente, etc.)		X	X
Verificar que las validaciones del lado del cliente se utilizan como una segunda línea de defensa, en adición a la validación del lado del servidor.		X	X
Verificar que todos los datos de entrada sean validados, no solamente los campos de formularios HTML sino también todos los orígenes de entrada como las llamadas REST, parámetros de consulta, encabezados HTTP, cookies, archivos por lotes, fuentes RSS, etc. Mediante validación positiva (lista blanca), o utilizando otras formas de validación menos eficaces tales como listas de rechazo transitorio (eliminando símbolos defectuosos), o rechazando malas entradas (listas negras)		X	X
Verificar que datos estructurados fuertemente tipados son validados con un esquema definido incluyendo; caracteres permitidos, longitud y patrones (p. ej. tarjeta de crédito o teléfono o validando que dos campos relacionados son razonables, tales como validación de coincidencia entre localidad y código postal).		X	X
Verificar que los datos no estructurados sean sanitizados cumpliendo medidas genéricas de seguridad tales como caracteres permitidos, longitud y que caracteres potencialmente dañinos en cierto contexto sean anulados (p. ej. nombres naturales con Unicode o apóstrofos, como ねこ o O'Hara).		X	X
Verificar que HTML no confiable proveniente de editores ***WYSIWYG*** o similares sean debidamente sanitizados con un sanitizador de HTML y se manejen apropiadamente según la validación de entrada y codificación.	X	X	X
Para tecnologías de plantilla de codificación automática, si ésta se ha deshabilitado, asegurar que la sanitización de HTML esté habilitada en su lugar.		X	X
Verificar que los datos transferidos desde un contexto DOM a otro, utilice métodos de JavaScript seguro, como pueden ser .innerText y .val		X	X
Verificar que cuando se interprete JSON en navegadores, que JSON.parse sea el utilizado para interpretarlo y no eval().		X	X
Verificar que los datos de autenticación se eliminen del almacenamiento del cliente, tales como el DOM del navegador después de terminada la sesión.		X	X