

# Seguridad en la Red Informática Mundial

## Top-Ten Vulnerabilidades según OWASP

Semana 10 clases 19 y 20

Mtra. María Noemí Araiza Ramírez

---

# Top-Ten Vulnerabilidades según OWASP



## Objetivos:

¿Qué es OWASP Top 10?

¿Qué objetivos y motivaciones tiene?

¿Qué ha cambiado de 2013 a 2017?

¿Qué son los Riesgos de Seguridad de Aplicaciones?

¿Conocer cómo me afectan o cuál es mi riesgo?

¿Conocer los 10 Riesgos más importantes?

---

# Top-Ten Vulnerabilidades según OWASP



**El software inseguro debilita:**

Finanzas

Salud

Defensa

Energía

Otras infraestructuras críticas

A medida que la infraestructura digital se hace cada vez más compleja e interconectada, la dificultad de lograr la seguridad en aplicaciones aumenta exponencialmente

---

# Top-Ten Vulnerabilidades según OWASP

## ¿Qué es TOP 10?

Representa una lista concisa y enfocada sobre los **Diez Riesgos Más Críticos sobre Seguridad en Aplicaciones**

El OWASP Top10 se centra en los riesgos

También provee información adicional sobre como evaluar estos riesgos en sus aplicaciones.

Cada ítem en el Top10 describe la probabilidad general y los factores de consecuencia que se utilizan para clasificar la gravedad típica del riesgo.

Presenta orientación sobre como verificar si usted posee problemas en esta área, como evitarlos, algunos ejemplos y enlaces a mayor información.

# Top-Ten Vulnerabilidades según OWASP



## Objetivos del Top 10:

Educar a desarrolladores, diseñadores, arquitectos, gerentes, y organizaciones sobre las consecuencias de las vulnerabilidades de seguridad más importantes en aplicaciones web.

Pretende crear conciencia sobre la seguridad en aplicaciones mediante la identificación de algunos de los riesgos más críticos que enfrentan las Organizaciones.

Top10 no es un programa de seguridad en aplicaciones.

---

# Top-Ten Vulnerabilidades según OWASP



## Objetivos del Top 10:

Referenciado por numerosos estándares, libros, y organizaciones, incluyendo MITRE, PCI DSS, DISA, FTC, y muchos más.

El OWASP Top 10 fue lanzado por primera vez en 2003, se han realizado actualizaciones posteriores en 2004, 2007, 2010, 2013 y 2017.

OWASP recomienda que las organizaciones establezcan una base sólida de formación, estándares y herramientas que hagan posible la codificación segura.

---

# Top-Ten Vulnerabilidades según OWASP



## Objetivos del Top 10:

Por encima de esa base, las organizaciones deben integrar la seguridad en su desarrollo, verificación y procesos de mantenimiento.

La gerencia puede utilizar los datos generados por estas actividades para gestionar los costos y riesgos asociados a la seguridad en aplicaciones.

---

# Top-Ten Vulnerabilidades según OWASP



## Objetivos del Top 10:

Se basa en el envío de datos de más de 40 empresas que se especializan en seguridad de aplicaciones y una encuesta de la industria que fue completada por más de 500 personas.

Esta información abarca vulnerabilidades recopiladas de cientos de organizaciones y más de 100.000 aplicaciones y APIs del mundo real.

Las 10 principales categorías fueron seleccionadas y priorizadas de acuerdo con estos datos de prevalencia, en combinación con estimaciones consensuadas de detectabilidad e impacto.

---



# Top-Ten Vulnerabilidades según OWASP



## ¿Qué ha cambiado de 2013 a 2017?

Ha habido cambios en los últimos cuatro años debido a que OWASP Top 10 necesitaba una actualización.

Se rediseñó y mejoró la metodología, así como, el proceso de obtención de datos, se trabajó con la comunidad, reordenaron los riesgos y se reescribieron desde cero, también se agregaron referencias a frameworks y lenguajes que son utilizados actualmente.

---

## ¿Qué ha cambiado de 2013 a 2017?

Nuevos riesgos, respaldados en datos:

A4:2017 – Entidades Externas XML (XXE) es una nueva categoría, respaldada principalmente por los resultados obtenidos de las herramientas de análisis estático de código (SAST).

---

# Top-Ten Vulnerabilidades según OWASP



## ¿Qué ha cambiado de 2013 a 2017?

Nuevos riesgos, respaldados por la comunidad:

Le pedimos a la comunidad que nos proporcionara información sobre dos categorías de debilidades. Luego de más de 500 envíos, y de eliminar los problemas que ya estaban respaldados por datos (tales como Exposición a Datos Sensibles y XXE), los dos nuevos riesgos son:

A8:2017 – Deserialización Insegura, que permite la ejecución remota de código o la manipulación de objetos sensibles en la plataforma afectada.

---

# Top-Ten Vulnerabilidades según OWASP



## ¿Qué ha cambiado de 2013 a 2017?

A10:2017 – Registro y Monitoreo Insuficientes, la falta de estos aspectos puede impedir o demorar en forma significativa la detección de actividad maliciosa o de sustracción de datos, la respuesta a los incidentes y la investigación forense digital.

---

# Top-Ten Vulnerabilidades según OWASP



## ¿Qué ha cambiado de 2013 a 2017?

Fusionados o retirados, pero no olvidados:

A4 – Referencia Directa Insegura a Objetos y A7 – Ausencia de Control de Acceso a las Funciones fueron fusionados en A5:2017 – Pérdida de Control de Acceso.

A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF) dado que varios Frameworks incluyen defensas contra CSRF, sólo se encontró en el 5% de las aplicaciones.

---

## ¿Qué ha cambiado de 2013 a 2017?

A10 – Redirecciones y reenvíos no validados, mientras que se encuentra en aproximadamente el 8% de las aplicaciones, fue superado ampliamente por XXE.

---

OWASP Top 10 2013	±	OWASP Top 10 2017
A1 – Inyección	→	A1:2017 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones	→	A2:2017 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	↘	A3:2017 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos [Unido+A7]	U	A4:2017 – Entidad Externa de XML (XXE) [NUEVO]
A5 – Configuración de Seguridad Incorrecta	↘	A5:2017 – Pérdida de Control de Acceso [Unido]
A6 – Exposición de Datos Sensibles	↗	A6:2017 – Configuración de Seguridad Incorrecta
A7 – Ausencia de Control de Acceso a las Funciones [Unido+A4]	U	A7:2017 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	✗	A8:2017 – Deserialización Insegura [NUEVO, Comunidad]
A9 – Uso de Componentes con Vulnerabilidades Conocidas	→	A9:2017 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados	✗	A10:2017 – Registro y Monitoreo Insuficientes [NUEVO, Comunidad]

# Top-Ten Vulnerabilidades según OWASP



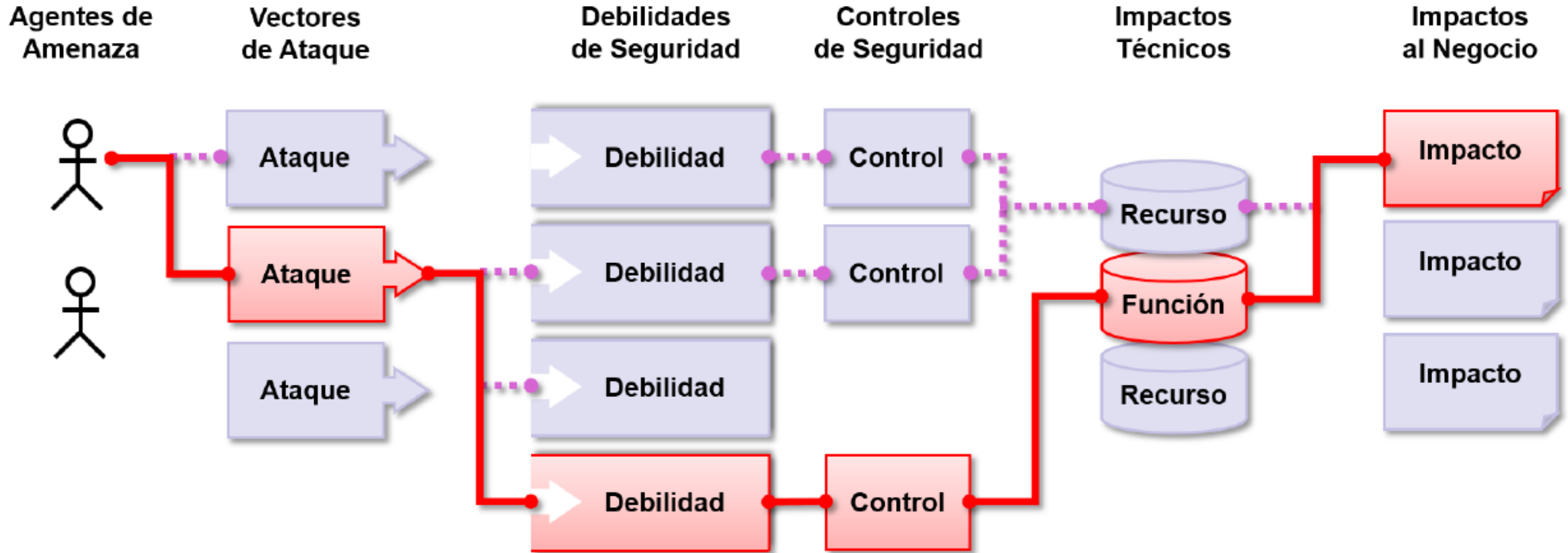
## **¿Cuáles son los riesgos en seguridad de aplicaciones?**

Los atacantes pueden, potencialmente, utilizar diferentes rutas a través de su aplicación para perjudicar su negocio u organización.

Cada uno de estos caminos representa un riesgo que puede o no ser suficientemente grave como para merecer atención.



## ¿Cuáles son los riesgos en seguridad de aplicaciones?



# Top-Ten Vulnerabilidades según OWASP



## **¿Cuáles son los riesgos en seguridad de aplicaciones?**

Algunas veces, estos caminos son fáciles de encontrar y explotar, mientras que otras son extremadamente difíciles.

De la misma manera, el perjuicio ocasionado puede no tener consecuencias, o puede dejarlo en la quiebra.

A fin de determinar el riesgo para su organización, puede evaluar la probabilidad asociada a cada agente de amenaza, vector de ataque, debilidad de seguridad y combinarlo con una estimación del impacto técnico y de negocio para su organización. Juntos, estos factores determinan su riesgo general.

---

# Top-Ten Vulnerabilidades según OWASP



## ¿Cuál es mi Riesgo?

El OWASP Top 10 se enfoca en identificar los riesgos más críticos para un amplio tipo de organizaciones.

Para cada uno de estos riesgos, se proporciona información genérica sobre la probabilidad y el impacto técnico, utilizando el siguiente esquema de evaluación, basado en la Metodología de Evaluación de Riesgos de OWASP.

---

# Top-Ten Vulnerabilidades según OWASP



## ¿Cuál es mi Riesgo?

Top 10 2013

Agente de Amenaza	Vectores de Ataque	Prevalencia de Debilidades	Detectabilidad de Debilidades	Impacto Técnico	Impacto al Negocio
Específico de la aplicación	Fácil	Difundido	Fácil	Severo	Específico de la aplicación /negocio
	Promedio	Común	Promedio	Moderado	
	Difícil	Poco Común	Difícil	Menor	

Agente de Amenaza	Explotabilidad	Prevalencia de Vulnerabilidad	Detección de Vulnerabilidad	Impacto Técnico	Impacto de Negocio
Específico de la Aplicación	Fácil 3	Difundido 3	Fácil 3	Severo 3	Específico del Negocio
	Promedio 2	Común 2	Promedio 2	Moderado 2	
	Difícil 1	Poco Común 1	Difícil 1	Mínimo 1	

Top 10 2017

# Top-Ten Vulnerabilidades según OWASP



## ¿Cuál es mi Riesgo?

Cada quien sabe los detalles específicos de su ambiente y su negocio.

Para una aplicación cualquiera, puede no haber un agente de amenaza que pueda ejecutar el ataque relevante, o el impacto técnico puede no ser relevante.

Por tanto, se debería evaluar cada riesgo, enfocándose en los agentes de amenaza, los controles de seguridad e impactos de negocio en su empresa.

Los nombres de los riesgos en el Top 10 surgen del tipo de ataque, el tipo de debilidad o el tipo de impacto que pueden causar.

---

# Top-Ten Vulnerabilidades según OWASP



## Tabla de amenaza

### A1- Inyección

Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un interprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al interprete en ejecutar comandos no intencionados o acceder datos no autorizados.

### A2 – Pérdida de Autenticación y Gestión de Sesiones

Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

# Top-Ten Vulnerabilidades según OWASP



## Tabla de amenaza

### **A3 – Secuencia de Comandos en Sitios Cruzados (XSS)**

Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

### **A4 – Referencia Directa Insegura a Objetos**

Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.



# Top-Ten Vulnerabilidades según OWASP



## Tabla de amenaza

### A5 – Configuración de Seguridad Incorrecta

Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.

### A6 – Exposición de datos sensibles

Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.



# Top-Ten Vulnerabilidades según OWASP



## Tabla de amenaza

### **A7 – Ausencia de Control de Acceso a Funciones**

La mayoría de aplicaciones web verifican los derechos de acceso a nivel de función antes de hacer visible en la misma interfaz de usuario. A pesar de esto, las aplicaciones necesitan verificar el control de acceso en el servidor cuando se accede a cada función. Si las solicitudes de acceso no se verifican, los atacantes podrán realizar peticiones sin la autorización apropiada.

### **A8 - Falsificación de Peticiones en Sitios Cruzados (CSRF)**

Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.

# Top-Ten Vulnerabilidades según OWASP



## Tabla de amenaza

### **A9 – Utilización de componentes con vulnerabilidades conocidas**



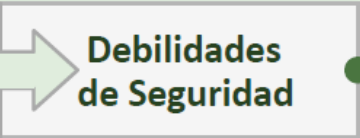


Algunos componentes tales como las librerías, los frameworks y otros módulos de software casi siempre funcionan con todos los privilegios. Si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una pérdida seria de datos. Las aplicaciones que utilicen componentes con vulnerabilidades conocidas debilitan las defensas de la aplicación y permiten ampliar el rango de posibles ataques e impactos.

### **A10 – Redirecciones y reenvíos no validados**

Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.

# Top-Ten Vulnerabilidades según OWASP

## A1 Inyección

 Agentes de Amenaza	 Vectores de Ataque	 Debilidades de Seguridad		 Impactos Técnicos	 Impactos al negocio
Específico de la Aplicación	Explotabilidad FÁCIL	Prevalencia COMÚN	Detección PROMEDIO	Impacto SEVERO	Específico de la aplicación/negocio
<p>Considere a cualquiera que pueda enviar información no confiable al sistema, incluyendo usuarios externos, usuarios internos y administradores.</p>	<p>El atacante envía ataques con cadenas simples de texto, los cuales explotan la sintaxis del interprete a vulnerar. Casi cualquier fuente de datos puede ser un vector de inyección, incluyendo las fuentes internas.</p>	<p>Las <u>fallas de inyección</u> ocurren cuando una aplicación envía información no confiable a un interprete. Estas fallas son muy comunes, particularmente en el código antiguo. Se encuentran, frecuentemente, en las consultas SQL, LDAP, Xpath o NoSQL; los comandos de SO, intérpretes de XML, encabezados de SMTP, argumentos de programas, etc. Estas fallas son fáciles de descubrir al examinar el código, pero difíciles de descubrir por medio de pruebas. Los analizadores y «fuzzers» pueden ayudar a los atacantes a encontrar fallas de inyección.</p>		<p>Una inyección puede causar pérdida o corrupción de datos, pérdida de responsabilidad, o negación de acceso. Algunas veces, una inyección puede llevar a el compromiso total de el servidor.</p>	<p>Considere el valor de negocio de los datos afectados y la plataforma sobre la que corre el intérprete. Todos los datos pueden ser robados, modificados o eliminados. ¿Podría ser dañada su reputación?</p>

# Top-Ten Vulnerabilidades según OWASP



**A1**

Inyección

## ¿Soy vulnerable?

Verificar que todo uso de intérpretes separe datos no confiables del comando o consulta. Para consultas SQL, esto significa utilizar variables parametrizadas en todas las consultas preparadas y procedimientos almacenados, como así también evitar consultas dinámicas.

Revisar el código es una manera fácil y efectiva para ver si la aplicación utiliza los interpretes de manera segura.

---

# Top-Ten Vulnerabilidades según OWASP



**A1**

**Inyección**

## ¿Soy vulnerable?

Las herramientas de análisis de código (SoapUI) pueden ayudar a un analista de seguridad a encontrar la utilización de interpretes y rastrear el flujo de datos en la aplicación.

Los test de penetración pueden validar estos problemas a través de fallas especialmente hechas a mano que confirman la vulnerabilidad.

---

# Top-Ten Vulnerabilidades según OWASP



**A1**

Inyección

## ¿Soy vulnerable?

Los escaneos dinámicos automatizados contra la aplicación pueden proveer una buena comprensión sobre si existe algún fallo que haga vulnerable mi aplicación a un ataque de inyección.

Los escáneres no siempre pueden llegar a los interpretes y tienen dificultad en detectar si un ataque ha tenido éxito. Una gestión pobre de los errores hace mas fácil la detección de fallos de inyección.

---

**A1**

**Inyección**

## ¿Soy vulnerable?

Todas las plataformas de aplicación Web que usen intérpretes o invoquen otros procesos son vulnerables a las fallas de inyección.

Esto incluye cualquier componente del marco de trabajo, que pueda usar intérpretes en la capa de bases de datos.

El objetivo es verificar que los datos del usuario no puedan modificar el significado de las órdenes y las consultas enviadas a ninguno de los intérpretes invocados por la aplicación.

---



## **A1** Inyección

### ¿Cómo se puede evitar?

Prevenir la inyección requiere mantener los datos no confiables separados de comandos y consultas.

Nos da tres alternativas:

1. La opción preferida es usar una API segura que evite el uso del intérprete o provea una interfaz parametrizada Hay que ser cuidadoso con API, tales como procedimientos almacenados, que son parametrizados, ya que aún pueden introducir inyección implícitamente Utilice API de confianza
-



# Top-Ten Vulnerabilidades según OWASP



## **A1** Inyección

### ¿Cómo se puede evitar?

2. Si no se encuentra disponible una API parametrizada, se deben eliminar los caracteres especiales, utilizando una sintaxis de escape especial para dicho intérprete OWASP's ESAPI posee algunas de estas rutinas de codificación.
  3. Una validación positiva de entradas con una apropiada canonicalización (proceso de selección de URL para un conjunto de contenido específico) es también recomendada, pero no es una defensa completa porque muchas aplicaciones requieren caracteres especiales en sus entradas OWASP's ESAPI tiene una librería extensible de rutinas de validación de entradas para Java.
-

**A1**

**Inyección**

## ¿Cómo se puede evitar?

Validación de entrada: Usar un mecanismo estándar de validación de entradas para validar todas las entradas contra el tamaño, el tipo, la sintaxis y las reglas de negocio antes de aceptar los datos que se van a mostrar o almacenar.

Evitar los mensajes de error detallados que son útiles para un atacante. Conceder los mínimos privilegios cuando se conecte a bases de datos y otros sistemas de bases de datos.

---

# Top-Ten Vulnerabilidades según OWASP



## **A1** Inyección

### Ejemplos de escenarios de ataques

#### Escenario No. 1

La aplicación usa datos no confiables en la construcción de la siguiente instrucción SQL vulnerable:

```
String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";
```

## **A1** Inyección

### Ejemplos de escenarios de ataques

#### Escenario No. 2

De forma similar, si una aplicación confía ciegamente en el framework puede resultar en consultas que aún son vulnerables, (ej Hibernate Query Language (HQL)):

```
Query HQLQuery = session.createQuery("FROM accounts WHERE custID='" + request.getParameter("id")  
+ "'");
```

## **A1** Inyección

### **Ejemplos de escenarios de ataques**

En ambos casos el atacante modifica el parámetro 'id' en su navegador para enviar: 'or '1'='1.

Por ejemplo: `http://example.com/app/accountView?id='or '1'='1`

Esto cambia el significado de la ambas consultas, regresando todos los registros de la tabla accounts en vez de sólo el cliente solicitado.

Ataques más peligrosos pueden modificar datos o incluso invocar procedimientos almacenados.

---

# Top-Ten Vulnerabilidades según OWASP

**A1**

Inyección

## Ejemplos de escenarios de ataques

